DFSMS/MVS Version 1 Release 5

General Information

GC26-4900-06

DFSMS/MVS Version 1 Release 5

General Information

GC26-4900-06

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page xi.

Seventh Edition (March 1999)

This edition applies to Version 1 Release 5 of DFSMS/MVS (5695-DF1), Version 2 Release 7 of OS/390 (5647-A01), and any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
RCF Processing Department
G26/050
5600 Cottle Road
SAN JOSE, CA 95193-0001
U.S.A.

Or you can send your comments electronically to starpubs@vnet.ibm.com.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1980, 1999. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 1980, 1999

Figures

© Copyright IBM Corp. 1980, 1999

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Information Enabling Requests
Dept. DWZ
5600 Cottle Road
San Jose, CA 95193

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Any pointers in this publication to non-IBM Web sites are provided for convenience only, and do not in any manner serve as an endorsement of these Web sites.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

© Copyright IBM Corp. 1980, 1999

AFP
AIX
AIX/ESA
AS/400
AT
BookManager
CICS
CICS/ESA
CICS/MVS
CUA
DATABASE 2
DB2
DFSMS
DFSMS/MVS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT
ES/9000
ESA/390
ESCON
GDDM
Hiperspace
IBM
IMS
IMS/ESA
MVS
MVS/DFP
MVS/ESA
MVS/SP
OS/2
OS/390
OS/400
Parallel Sysplex
Print Services Facility
QMF
RACF
RAMAC
RETAIN
RISC System/6000
RMF
RS/6000
S/370
S/390
System/36
System/38
System/370
System/390
VM/ESA
VSE/ESA

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Windows is a trademark of Microsoft Corporation.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Summary of Changes

This summary of changes includes specific updates to this book as well as a history of the updates made in previous editions of this book.

Seventh Edition, March 1999

This publication is a major revision in support of the technical changes introduced with DFSMS/MVS Version 1 Release 5. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This revision also includes maintenance and editorial changes.

The following technical changes have been made to this publication:

- Changed the OpenEdition/MVS product name to OS/390 UNIX System Services.
- Removed DFSMS/MVS Network File System section from Chapter 1, Introduction to DFSMS/MVS.
- Added Internet addresses for the DFSMSdss, DFSMShsm, and DFSMSrmm home pages in Figure 3 on page 10, in Chapter 1, Introduction to DFSMS/MVS.
- Revised the chapter, Chapter 2, “What's New in DFSMS/MVS Version 1 Release 5?” on page 11, which summarizes each enhancement of the DFSMS/MVS 1.5 release.
- Added a new section on Virtual Tape Servers, Devices, and Volumes in Chapter 3, An Overview of Storage Management With DFSMS/MVS.
- Expanded list of storage administrator tasks that can be defined through data class in “Managing Data Set Attributes” on page 24, in Chapter 4, Storage Management with DFSMSdfp.
- Updated the list of storage administrator tasks that can be defined through data class to include system-managed buffering for all VSAM data set types in “Managing Data Set Attributes” on page 24, in Chapter 4, Storage Management with DFSMSdfp.
- Updated the sample ISMF primary option menus in “Interactive Storage Management Facility and SMS-Managed Storage” on page 37, in Chapter 4, Storage Management with DFSMSdfp.
- Updated the list of NaviQuest capabilities in Chapter 4, Storage Management with DFSMSdfp.
- Updated the list of things that can be done with extended-format data sets in Chapter 5, Data Management with DFSMSdfp.
- Updated "Selecting Space Allocations on New Volumes" in Chapter 5, Data Management with DFSMSdfp.
- Updated information in the section "Extended Addressability" in Chapter 5, Data Management with DFSMSdfp.
- Updated information in the section "Using System-Managed Buffering for VSAM Batch Programs" in Chapter 5, Data Management with DFSMSdfp.

© Copyright IBM Corp. 1980, 1999

- Added information on the data management utility IEHINITT in “Data Management Utilities” on page 46, in Chapter 5, Data Management with DFSMSdfp.
- Renamed section on "Character Data Representation" to “Character Data Representation and Data Conversion” on page 47 in Chapter 5, Data Management with DFSMSdfp.
- Added information on data conversion using ISO/ANSI Version 4 tapes to section on “Character Data Representation and Data Conversion” on page 47 in Chapter 5, Data Management with DFSMSdfp.
- Added a new section describing enhancements to the catalog sharing capability in Chapter 5, Data Management with DFSMSdfp.
- Added information on CICSVR as a data recovery tool in “Using VSAM Record-Level Sharing” on page 55 in Chapter 5, Data Management with DFSMSdfp.
- Renamed “Common Supported IBM I/O Devices” on page 69 in Chapter 7, Device Management with DFSMSdfp.
- Added 3494 Virtual Tape Server to list of supported devices in “Common Supported IBM I/O Devices” on page 69.
- Removed section on the "Network File System" in Chapter 8, Distributed Data Access with DFSMSdfp.
- Added a new section on DFSMSdss support of RVA Snapshot in Chapter 9, Role of the Functional Component DFSMSdss.

Note: For other important updates to this book, please check informational APAR II11474, a repository of DFSMS/MVS 1.5 information that was not available at the time DFSMS/MVS books were published for general availability.

Sixth Edition, June 1997

This publication is a major revision in support of the technical changes introduced with DFSMS/MVS Version 1 Release 4. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This revision also includes maintenance and editorial changes.

The following summarizes changes to that information.

- Added internet addresses for the DFSMS/MVS home page, as well as for DFSMS/MVS features and supporting products, in “Product Offerings and Distribution” on page 7, in Chapter 1, Introduction to DFSMS/MVS.
- Added a new chapter, Chapter 2, “What's New in DFSMS/MVS Version 1 Release 5?” on page 11, which summarizes each enhancement of the DFSMS/MVS 1.4 release.
- Added information about moving your peer-to-peer remote copy (PPRC) data easily using PPRC dynamic address switching (P/DAS), in “Move Your Peer-to-Peer Remote Copy (PPRC) Data Easily with P/DAS” on page 20, in Chapter 3, “An Overview of Storage Management With DFSMS/MVS” on page 15.
- Added information about new data class attributes, in “Managing Data Set Attributes” on page 24, in Chapter 4, Storage Management with DFSMSdfp.

- Added information about altering the storage or management class of a migrated data set, in “Using SMS Classes and Groups” on page 23, in Chapter 4, Storage Management with DFSMSdfp.
- Updated the sample ISMF primary option menus in “Interactive Storage Management Facility and SMS-Managed Storage” on page 37, in Chapter 4, Storage Management with DFSMSdfp.
- Added a new section describing using the NaviQuest tool in batch, in Chapter 4, Storage Management with DFSMSdfp.
- Added information on tailored compression and system-managed buffering, in “Creating and Maintaining Data Sets” on page 41, in Chapter 5, Data Management with DFSMSdfp.
- Added information about a new DFSMSdfp callable service, IGWABWO, which you can use to back up CICS VSAM File Control data sets while they are open for update, in “DFSMSdfp Callable Services” on page 53, in Chapter 5, Data Management with DFSMSdfp.
- Added information about enhancements to the program management binder and loader in Chapter 6, “Program Management with DFSMSdfp” on page 61, including information on dynamic link libraries (DLLs) and a new section on loading modules in a shared environment.
- Updated Figure 13 on page 69, the table of common IBM I/O devices supported by MVS/ESA SP and OS/390, in Chapter 7, “Device Management with DFSMSdfp” on page 69.
- Added a new section describing Distributed FileManager/MVS DataAgent.
- Added information about the improved DFSMSdss Stand-Alone data recovery solution, in “DFSMSdss Stand-Alone Services” on page 88, in Chapter 9, Role of the Functional Component DFSMSdss.
- Updated and expanded the glossary, index, and library appendix entries.

Fifth Edition, December 1995

This publication is a minor revision in support of the technical changes introduced with DFSMS/MVS Version 1 Release 3. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This revision also includes maintenance and editorial changes.

The following summarizes the changes to that information.

- Created a new chapter, Chapter 3, “An Overview of Storage Management With DFSMS/MVS” on page 15, from information formerly included in Chapter 1, “Introduction to DFSMS/MVS” on page 1.
- Added additional pointers to publications with information on VSAM record-level sharing.
- Updated and expanded the section on functional components and features, to discuss related products.
- Updated and expanded the glossary, index, and DFSMS/MVS library appendix entries.

Fourth Edition, May 1995

This publication is a major revision in support of the technical changes introduced with DFSMS/MVS Version 1 Release 3. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This revision also includes maintenance and editorial changes.

The following summarizes the changes to that information.

- Added Chapter 1, "Introduction to DFSMS/MVS" on page 1, a new introduction.
- Added, Appendix A, "DFSMS/MVS Library and Related Publications" on page 129 a new section that consolidates the lists of DFSMS/MVS books and related product publications into a single reference. The table containing DFSMS/MVS publications includes audience information for each book.
- Added the following new sections describing VSAM record-level sharing (RLS):
  - "Defining Use of the Coupling Facility for VSAM Record-Level Sharing" on page 35 in Chapter 4, Storage Management with DFSMSdfp
  - "Using VSAM Record-Level Sharing" on page 55 in Chapter 5, Data Management with DFSMSdfp
- Added information about 32-name support to "Defining System Group Name and 32-Name Support" on page 34 in Chapter 4, Storage Management with DFSMSdfp
- Added information about using extended addressability for system-managed key-sequenced data sets allocated in extended format, to support data sets that contain more than 4 gigabytes (GB) of data, to "Extended Format Data Sets" on page 43 in Chapter 5, Data Management with DFSMSdfp.
- Created a new section describing Character Data Representation Architecture (CDRA): "Character Data Representation and Data Conversion" on page 47 in Chapter 5, Data Management with DFSMSdfp.
- Added information about Storage Management Subsystem (SMS) checking without using the RESOWNER value to "Resource Access Control Facility Protection" on page 57 in Chapter 5, Data Management with DFSMSdfp.
- Created a new section describing extended remote copy (XRC): "Disaster Recovery and Application Migration" on page 52 in Chapter 5, Data Management with DFSMSdfp.
- Added information about a new DFSMSdfp callable service, IGWARLS, to "DFSMSdfp Callable Services" on page 53 in Chapter 5, Data Management with DFSMSdfp.
- Updated Chapter 6, "Program Management with DFSMSdfp" on page 61 with information about program management extensions for DFSMS/MVS 1.3.
- Updated the table of common IBM I/O devices supported by MVS, and moved the table to Chapter 7, "Device Management with DFSMSdfp" on page 69.
- Updated all ISMF panels.
- Updated existing index and glossary entries, and added new entries.

About This Book

This book is intended to help you learn about and evaluate the DFSMS/MVS Version 1 Release 5 licensed program. Anyone with a general background in programming and familiar with operating systems can use this book as an overview of DFSMS/MVS and its components. In addition, system programmers planning to install DFSMS/MVS can use this book to learn about DFSMS/MVS in the MVS/ESA SP and OS/390 environments. DFSMS/MVS works with MVS/ESA SP Version 5 and with OS/390.

DFSMS/MVS Version 1 Release 5 introduces enhancements to the evolving system-managed storage environment. System-managed storage uses a combination of hardware, software, and installation-defined policies to automatically perform storage tasks that storage administrators and users would otherwise need to perform manually.

How to Tell if this Book is Current

IBM regularly updates its books with new and changed information. When first published, both hardcopy and BookManager softcopy versions of a book are identical, but subsequent updates might be available in softcopy before they are available in hardcopy. Here's how to determine the level of a book:

- Check the book's order number suffix (often referred to as the dash level). A book with a higher dash level is more current than one with a lower dash level. For example, in the publication order number SC26-4930-02, the dash level 02 means that the book is more current than previous levels, such as 01 or 00. Suffix numbers are updated as a product moves from release to release, as well as for hardcopy updates within a given release.
- Check to see if you are using the latest softcopy version. To do this, compare the last two characters of the book's file name (also called the book name). The higher the number, the more recent the book. For example, DGT1U302 is more recent than DGT1U301.
- Compare the dates of the hardcopy and softcopy versions of the books. Even if the hardcopy and softcopy versions of the book have the same dash level, the softcopy could be more current. This will not be apparent from looking at the edition notice. The edition notice number and date remain that of the last hardcopy version. When you are looking at the softcopy product bookshelf, check the date shown to the right of the book title. This will be the date that the softcopy version was created.

  Also, an asterisk (*) is added next to the new and changed book titles in the CD-ROM booklet and the README files.

Vertical lines to the left of the text indicate changes or additions to the text and illustrations. For a book that has been updated in softcopy only, the vertical lines indicate changes made since the last printed version.

© Copyright IBM Corp. 1980, 1999

Referenced Publications

Within the text, references are made to other DFSMS/MVS books, as well as to books of related products. See Appendix A, "DFSMS/MVS Library and Related Publications" on page 129 for the titles and order numbers for the DFSMS/MVS books, the Storage Subsystem Library, the general information manuals of related products, and other books referred to in this publication. The titles and order numbers of referenced publications in the OS/390 library are listed in the *OS/390 Information Roadmap*.

This book also references several publications produced by the International Technical Support Organization (ITSO). ITSO books are often called *redbooks*. The information contained in these redbooks has not been submitted to any formal IBM test and is distributed *as is*. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment.

References to Product Names Used in DFSMS/MVS Publications

DFSMS/MVS publications support DFSMS/MVS, 5695-DF1, as well as the DFSMSdfp base element and the DFSMShsm, DFSMSdss, and DFSMSrmm features of OS/390, 5647-A01. DFSMS/MVS publications also describe how DFSMS/MVS interacts with other IBM products to perform the essential data, storage, program and device management functions of the operating system.

DFSMS/MVS publications typically refer to another IBM product using a generic name for the product. When a particular release level of a product is relevant, the reference includes the complete name of that product. This section explains the naming conventions used in the DFSMS/MVS library for the following products:

MVS can refer to:

- MVS/ESA SP Version 5, 5695-047 or 5695-048
- The MVS base control program (BCP) of OS/390, 5647-A01

All MVS book titles used in DFSMS/MVS publications refer to the OS/390 editions. Users of MVS/ESA SP Version 5 should use the corresponding MVS/ESA book. Refer to *OS/390 Information Roadmap* for titles and order numbers for all the elements and features of OS/390.

For more information about OS/390 elements and features, including their relationship to MVS/ESA SP and related products, please refer to *OS/390 Planning for Installation*.

RACF can refer to:

- Resource Access Control Facility (RACF), Version 2, 5695-039
- The RACF element of the OS/390 Security Server, an optional feature of OS/390

All RACF book titles refer to the Security Server editions. Users of RACF Version 2 should use the corresponding book for their level of the product. Refer to *OS/390 Security Server (RACF) Introduction* for more information about the Security Server.

CICS can refer to:

- CICS/MVS, 5665-403
- CICS/ESA, 5685-083
- The CICS element of the CICS Transaction Server for OS/390, 5665-147

All CICS book titles refer to the CICS Transaction Server for OS/390 editions. Users of CICS/MVS and CICS/ESA should use the corresponding books for those products. Please see *CICS Transaction Server for OS/390: Planning for Installation* for more information.

Chapter 1. Introduction to DFSMS/MVS

This chapter introduces DFSMS/MVS and its functional components. It briefly discusses how DFSMS/MVS can help you manage your storage, allowing you to better cope with your growing storage needs. It also introduces the available configurations of DFSMS/MVS.

See the announcement information for DFSMS/MVS Version 1 Release 5 and *DFSMS/MVS Program Directory*, shipped with the product, for information on DFSMS/MVS operating requirements. For complete information on planning to install DFSMS/MVS, see *DFSMS/MVS Planning for Installation*.

The DFSMS/MVS Storage Environment—Solving Your Growing Storage Needs

As your business expands, so do your needs for storage to hold your applications and data, and the costs of managing that storage. Storage costs include more than the price of the hardware, with the highest cost being the people needed to perform storage management tasks. If your business requires transaction systems, the batch window can also be a high cost. Additionally, you must pay for people to install, monitor, and operate your storage hardware devices, for electrical power to keep each piece of storage hardware cool and running, and for floor space to house them. Removable media, such as optical and tape storage, cost less per gigabyte (GB) than online storage, but require additional time and resources to locate, retrieve, and mount.

To allow your business to grow efficiently and profitably, you want to find ways to control the growth of your information systems and use your current storage more effectively.

The DFSMS/MVS software product, together with IBM hardware products, and your installation-specific requirements for data and resource management, comprise the key to system-managed storage in an MVS environment. The components of DFSMS/MVS automate and centralize storage management, based on policies your installation defines for availability, performance, space, and security. The Interactive Storage Management Facility (ISMF) provides the user interface for defining and maintaining these policies, which the Storage Management Subsystem (SMS) governs for the system.

DFSMS/MVS Functional Components

DFSMS/MVS is comprised of the following functional components:

DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm

DFSMS/MVS provides and enhances functions formerly provided by MVS/DFP, Data Facility Data Set Services (DFDSS), and the Data Facility Hierarchical Storage Manager (DFHSM). DFSMS/MVS is easier to install and order than those earlier offerings, as it eliminates the need to install multiple products. The tighter

© Copyright IBM Corp. 1980, 1999

integration of functional components in DFSMS/MVS and more integrated testing have also improved overall quality. More importantly, DFSMS/MVS improves upon earlier offerings by adding support for new functions.

The following sections list the services each functional component provides and direct you to specific chapters in this book for more information.

DFSMSdfp

DFSMSdfp provides the foundation for:

Storage management

DFSMSdfp includes ISMF, an interactive facility that lets you define and maintain policies to manage your storage resources. These policies help to improve the use of storage devices and to increase levels of service for user data, with minimal effort required from users. SMS manages these policies for the operating system. You can also use the NaviQuest tool under ISMF to help you migrate to SMS, maintain your SMS configuration, and perform many testing, implementation, and reporting tasks in batch.

See Chapter 4, “Storage Management with DFSMSdfp” on page 23 for more information.

Tape mount management

SMS provides a means for implementing tape mount management, a methodology for improving tape usage and reducing tape costs. This methodology involves intercepting selected tape data set allocations through the SMS automatic class selection (ACS) process and redirecting them to a DASD buffer. Once on DASD, these data sets can be migrated to a single tape or small set of tapes, thereby reducing the overhead associated with multiple tape mounts.

Data management

DFSMSdfp helps you store and catalog information on DASD, optical, and tape resources so that it can be quickly identified and retrieved from the system. You can use the Catalog Search Interface, now part of DFSMSdfp, to access the catalog.

See Chapter 5, “Data Management with DFSMSdfp” on page 41 for more information.

Program management

DFSMSdfp combines programs into executable modules, prepares them to run on the operating system, stores them in libraries, and reads them into storage for execution.

See Chapter 6, “Program Management with DFSMSdfp” on page 61 for more information.

Device management

DFSMSdfp is involved in defining your input and output devices to the system and in controlling the operation of those devices in the MVS/ESA environment.

See Chapter 7, “Device Management with DFSMSdfp” on page 69 for more information.

Distributed data access

Distributed data access allows all authorized systems and users in a network to exploit the powerful features of system-managed storage, or automated storage management provided by DFSMS/MVS. DFSMSdfp uses the Distributed

FileManager (DFM) to support remote access of MVS data and storage resources from workstations, personal computers, or any other system on a SNA LU 6.2 network.

| The Hierarchical File System (HFS) works in conjunction with OS/390 UNIX to provide a full UNIX environment within the MVS system. MVS becomes a full-feature UNIX client or server when coupled with the OS/390 Network File System (OS/390 NFS) or Network File System R4 (pre-OS/390 V2.6 level of Network File System). With HFS, MVS programs can directly access UNIX data. When the OS/390 NFS client and OS/390 UNIX are used together, MVS can act as a client and access data from any remote system, including another MVS or UNIX system that is connected using a TCP/IP network served by a Network File System server.

For more information on the OS/390 Network File System, see *OS/390 Network File System User's Guide*. For the specific requirements on coupling MVS with Network File System R4, see *DFSMS 1.3 Network File System Customization and Operations Guide*.

DFSMSdss

DFSMSdss is used for:

Data movement and replication
DFSMSdss lets you move or copy data between volumes of like and unlike device types. It can also copy data that has been backed up.

Space management
DFSMSdss can reduce or eliminate DASD free-space fragmentation.

Data backup and recovery
DFSMSdss provides you with host system backup and recovery functions at both the data set and volume levels. It also includes a stand-alone restore program that you can run without a host operating system.

Data set and volume conversion
DFSMSdss can be used to convert your data sets and volumes to system-managed storage. It can also return your data to a non-system-managed state as part of a recovery procedure.

See Chapter 9, “Role of the Functional Component DFSMSdss” on page 83 and Chapter 10, “DFSMSdss Facilities” on page 91 for more information.

DFSMShsm

DFSMShsm provides functions for:

Storage management
DFSMShsm uses a hierarchy of storage devices in its automatic management of data, relieving end-users from manual storage management tasks.

Space management
DFSMShsm improves DASD space usage by keeping only active data on fast-access storage devices. It automatically frees space on user volumes by deleting eligible data sets, releasing over-allocated space, and moving low-activity data to lower cost-per-byte devices.

Tape mount management
DFSMShsm can write multiple output data sets to a single tape, making it a useful tool for implementing the tape mount management methodology under

SMS. When you redirect tape data set allocations to DASD, DFSMShsm can move those data sets to tape, as a group, during interval migration. This greatly reduces the number of tape mounts on the system. DFSMShsm uses single file format, which improves your tape usage and search capabilities.

Availability management
DFSMShsm backs up your data—automatically or by command—to ensure availability in the event of accidental loss of the data sets or physical loss of volumes. DFSMShsm also allows the storage administrator to copy backup and migration tapes, and to specify that copies be made in parallel with the original. The copies can be stored on site as protection from media damage, or off site as protection from site damage. Disaster backup and recovery is also provided for user-defined groups of data sets (aggregates) so that critical applications can be restored at the same location or at an off-site location.

Full exploitation of DFSMShsm services in a DFSMS environment requires the use of DFSMSdss for certain functions. You can also use the DFSMS/MVS Optimizer feature to maximize your storage subsystem's performance and minimize the storage hierarchy's costs. The Optimizer feature can also monitor and tune multiple DFSMShsm address spaces in a Parallel Sysplex.

See Chapter 11, “Role of the Functional Component DFSMShsm” on page 95 and Chapter 12, “DFSMShsm Facilities” on page 117 for more information.

DFSMSrmm

DFSMSrmm manages your removable media resources, including tape cartridges and reels. It provides functions for:

Library Management
You can create tape libraries, or collections of tape media associated with tape drives, to balance the work of your tape drives and operators.

DFSMSrmm can manage:

- System-managed tape libraries, such as the IBM 3494 and IBM 3495 Automated Tape Library Dataservers, and the manual IBM 3495 Tape Library Dataserver Model M10.
- Non-system-managed, or traditional, tape libraries, including automated libraries which are not system-managed, such as the IBM 3494 and IBM 3495 Tape Library Dataservers under Basic Tape Library Support (BTLS) control.

Shelf Management
DFSMSrmm groups information about removable media by shelves into a central online inventory, and keeps track of the volumes residing on those shelves. DFSMSrmm can optionally provide shelf management for storage locations used to vault tapes outside of the tape library.

Volume management
DFSMSrmm helps to manage the movement and retention of tape volumes throughout their life cycle.

Data set management
DFSMSrmm records information about the data sets on tape volumes to validate volume and data set information and to help maintain data integrity. It can also control the retention of those data sets.

See Chapter 13, “Role of the Functional Component DFSMSrmm” on page 123 for more information.

Additional DFSMS/MVS Features and Supporting Products

The following separately orderable features and products are also available to complement DFSMS/MVS and its various functions:

DFSMS/MVS Optimizer Feature
Data Facility Sort (DFSORT)
Adstar Distributed Storage Manager (ADSM) for MVS Version 2

DFSMS/MVS Optimizer Feature

The DFSMS/MVS Optimizer Feature is a separately orderable feature of DFSMS/MVS that provides analysis and simulation information for both SMS and non-SMS data. The DFSMS/MVS Optimizer Feature can help you maximize storage use and minimize storage costs. It provides methods and facilities for you to:

- Monitor and tune DFSMShsm functions.
- Create and maintain a historical database of system and data activity.
- Perform in-depth analysis of:
  - management class policies, including simulations and cost-benefit analysis using your storage component costs.
  - storage class policies for SMS data, with recommendations for placement and simulation for cache and expanded storage.
  - high I/O activity data sets, including recommendations for placement and simulation for cache and expanded storage.
  - storage hardware performance of subsystems and volumes including I/O rate, response time, and caching statistics.
- Fine-tune an SMS configuration, by helping you:
  - understand how current SMS policies and procedures are managed.
  - determine associated costs of current data management practices.
  - simulate potential policy changes and understand the costs of those changes.
- Produce presentation-quality charts.

For more information on the DFSMS/MVS Optimizer Feature, see *DFSMS Optimizer User's Guide and Reference*, or the DFSMS/MVS Optimizer home page at:

```
http://www.storage.ibm.com/optimizer
```

Note that the DFSMS/MVS Optimizer feature is not an optional feature of OS/390; if you are an OS/390 customer, order the standalone Optimizer product (5655-OPT).

Data Facility Sort (DFSORT)

Data Facility Sort (DFSORT), a separately priced feature of OS/390, is a highly flexible data processing tool that provides fast and efficient sorting, merging, copying, reporting, and analysis of your business information. Following are a few of the many tasks you can perform with DFSORT:

- Sort, merge, or copy files; including or excluding records, and reformatting records.
- Analyze data and produce detailed reports using the ICETOOL utility or the OUTFIL function. In addition, OUTFIL allows you to create different views of the data and different reports with a single pass over the data.
- Adapt to the sorting and merging needs of different countries using DFSORT's national language support.
- Provide correct ordering of 2-digit years and transformation of 2-digit years to 4-digit years as part of IBM's year 2000 solution.

For more information on DFSORT, see *DFSORT Getting Started R14* or the DFSORT home page at:

```
http://www.storage.ibm.com/sort
```

DFSORT is a standalone product (5740-SM1) or an optional priced feature of OS/390 Release 2 and higher.

Adstar Distributed Storage Manager (ADSM) for MVS Version 2

ADSM for MVS Version 2 is a client/server storage management product that provides administrator-controlled, highly automated, centrally scheduled, network-based backup, archive and space management functions for workstations and LAN file servers. The ADSM client on MVS provides support at the file level for data residing on HFS.

With ADSM for MVS Version 2, you can:

- Automate backup and archive processing for distributed systems
- Provide hierarchical storage management for AIX workstations and servers, allowing you to migrate less-used file to an ADSM server
- Incrementally back up OS/390 UNIX file systems
- Centrally manage your diverse environment from an administrator interface on TSO, AIX, OS/2 and many other platforms

ADSM for MVS Version 2 has a broad range of support for desktop clients and file servers, including the following:

Apple Macintosh**	IBM OS/2 Warp	Siemens Nixdorf SINIX**
AT&T GIS SVR-4**	Microsoft LanManager	SiliconGraphics IRIX
BULL BOS/X	NEC EWS-UX/V	SunOS**
DEC Ultrix**	Novell Netware**	Sun Solaris**
DOS	OS/390 UNIX System Services	Windows
HP-UX**	SCO UNIX 386**	Windows for Workgroups
IBM AIX	SCO Open Desktop**	Windows NT
IBM OS/2 LanServer	Sequent DYNIX/ptx	

For more information on ADSM, see *ADSTAR Distributed Storage Manager: General Information* or the ADSM home page at:

```
http://www.storage.ibm.com/adsm
```

Product Offerings and Distribution

The full-function DFSMS/MVS product provides all four functional components: DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm. You can also choose from among several DFSMS/MVS alternate offerings. With an alternate offering, you obtain a license to activate only the DFSMS/MVS functional components that meet your business needs.

Available Configurations of DFSMS/MVS

Figure 1 shows the functional components you can use for each available configuration of DFSMS/MVS.

Figure 1. Configurations of DFSMS/MVS

Configuration	**Functional Components of DFSMS/MVS**
Full-Function Product	DFSMSdfp, DFSMSdss, DFSMShsm, DFSMSrmm
Alternate Offering 1	DFSMSdfp
Alternate Offering 2	DFSMSdfp, DFSMSdss
Alternate Offering 3	DFSMSdfp, DFSMSdss, DFSMShsm
Alternate Offering 4	DFSMSdfp, DFSMSrmm
Alternate Offering 5	DFSMSdfp, DFSMSdss, DFSMSrmm

Because of inter-component dependencies, you must install all four functional components shipped on the program tape, regardless of which option you choose. You are charged, however, for only the components you actually use.

Be aware that DFSMS/MVS cannot coexist with an earlier version of its component parts; for example, you cannot run with Alternate Offering 1 (DFSMSdfp) and DFHSM Version 2 Release 6 on the same system.

Figure 2 on page 8 lists the functions available for each configuration of DFSMS/MVS.

Figure 2. Available Functions for DFSMS/MVS Configurations

Functions Included	**Full Function Product**	**Alternate Offerings**				
		1	**2**	**3**	**4**	**5**
Data Management	X	X	X	X	X	X
Program Management	X	X	X	X	X	X
Device Management	X	X	X	X	X	X
Distributed Data Management (using Distributed FileManager)	X	X	X	X	X	X
Storage Management	X	X	X	X	X	X
Storage management policy definitions	X	X	X	X	X	X
Interactive panels	X	X	X	X	X	X
Data placement	X	X	X	X	X	X
Tape libraries, with or without automation	X	X	X	X	X	X
Object Access Method (OAM)	X	X	X	X	X	X
Hardware tape data compaction	X	X	X	X	X	X
Reports on device use	X	X	X	X	X	X
Reports on tape transports or mounts	X	X	X	X	X	X
Movement of data between DASD volumes	X		X	X		X
Backup by DASD volume	X		X	X		X
Non-disruptive (concurrent copy) backup or copy of data	X		X	X		X
Stand-alone restore of data	X		X	X		X
DASD data set statistics (ISMF/DCOLLECT)	X	X	X	X	X	X
Access method data compression	X	X	X	X	X	X
Automatic data migration or recall	X			X		
Software data compaction during migration	X			X		
Tape mount management	X			X		
Automated dump of DASD volumes	X			X		
Automatic deletion of expired or temporary data	X			X		
Automatic backup of new or changed data	X			X		
Aggregate backup and recovery	X			X		
Reports on removable media	X				X	X
Managing removable media volume, shelf space	X				X	X
Reusing tape volumes with expired data	X				X	X
UNIX support in MVS/ESA through the HFS	X	X	X	X	X	X
UNIX support in OS/390 through the HFS	X	X	X	X	X	X
Extended remote copy (XRC)	X	X	X	X	X	X
Peer-to-peer remote copy (PPRC)	X	X	X	X	X	X

Distribution Package

Each DFSMS/MVS license holder receives the following as part of the product distribution package:

- DFSMS/MVS program tapes
- Optional DFSMS/MVS feature tape
- *DFSMS/MVS Program Directory*
- Single printed copy of selected DFSMS/MVS publications (see Appendix A, “DFSMS/MVS Library and Related Publications” on page 129 for a complete list of DFSMS/MVS publications).
- The *IBM Online Library Omnibus Edition OS/390 Collection*, a CD-ROM containing all DFSMS/MVS publications for this release. Because it contains unencrypted licensed materials, this CD-ROM is available only to DFSMS/MVS licensees.

See your *DFSMS/MVS Program Directory* in the offering package for information about the program materials along with a complete list of requirements and instructions for installing DFSMS/MVS.

When you order any DFSMS/MVS feature, you receive a separate program directory containing instructions for installing those features on your system.

Additional Publications

Appendix A, “DFSMS/MVS Library and Related Publications” on page 129 contains a list of DFSMS/MVS and related publications, complete with task and audience descriptions, and order numbers. The *DFSMS/MVS Library Guide* describes the changes to content, structure, or titles incorporated into the latest release of the DFSMS/MVS library, to help you make the transition from a previous release. You can order additional copies of DFSMS/MVS publications through the IBM Distribution Center or through your IBM representative. Electronic access is also available.

You can order the following CD-ROMs to receive product libraries in BookManager-readable format:

- *IBM Online Library Omnibus Edition MVS Collection*, which contains MVS product libraries, including all of the DFSMS/MVS publications
- *IBM Online Library Omnibus Edition OS/390 Collection*

Internet Information about DFSMS/MVS

Information about DFSMS/MVS and its features is available on the Internet, at the following addresses:

Figure 3. DFSMS/MVS Home Page Addresses

Product or Feature	Address
DFSMS/MVS	http://www.storage.ibm.com/dfsms
DFSMSdss	http://www.storage.ibm.com/software/sms/dss/dsshome.htm
DFSMShsm	http://www.storage.ibm.com/software/sms/hsm/hsmhome.htm
DFSMSrmm	http://www.storage.ibm.com/software/rmm/rmmhome.htm
DFSORT	http://www.storage.ibm.com/sort
DFSMS/MVS Optimizer	http://www.storage.ibm.com/optimizer
ADSM	http://www.storage.ibm.com/adsm

Service Agreements

DFSMS/MVS is a licensed program supported by IBM Central Service which includes the IBM Support Center. Consult your IBM representative for information on the service agreement for DFSMS/MVS.

Chapter 2. What's New in DFSMS/MVS Version 1 Release 5?

With Release 5, DFSMS/MVS introduces enhancements to VSAM, catalog, OAM, NaviQuest, DFSMSrmm, and DFSMShsm. In addition, the performance of the Hierarchical File System has been improved. Release 5 also removes the limitation of 8191 open data sets for IMS DB and CICS VSAM, and adds support for ISO/ANSI Version 4 tapes. These and additional enhancements are briefly described in the following sections.

For a more complete description of the enhancements specific to Release 5, see *DFSMS/MVS Planning for Installation*.

VSAM Extended Format Data Set Enhancements

The VSAM extended-format data set enhancements include:

- **System-Managed Buffering**

  Provides the capability of having the system determine the number, size, and buffering algorithms to use for ESDS, RRDS, VRRDS, and LDSs.

- **Partial Release**

  Provides the capability of releasing allocated but unused space in ESDS, RRDS, VRRDS, and LDS. It is supported with a management class parameter and JCL.

- **Candidate Volume Space Amount**

  Provides the user with the choice of using either the primary space quantity or the secondary space quantity when a data set extends to a new volume.

- **Extended Addressability**

  Provides the ability for ESDS, RRDS, VRRDS, LDSs to grow to more than 4 gigabytes in size.

Enhanced Catalog Sharing

Catalog sharing enhancements include:

- **Catalog Sharing Performance**

  Performance of shared ICF catalogs is improved in a sysplex environment through the use of the S/390 Coupling Facility cache structure instead of DASD for the catalog control record.

Additional Enhancements

Additional enhancements in this release are described here:

- **Support for more than 8191 DDs for IMS DB and CICS VSAM**

  Allows full exploitation of the IMS 6.1 and CICS Transaction Server DD limit by expanding the number of data sets that can be opened and allocated to more than 10,000.

© Copyright IBM Corp. 1980, 1999

- **ISO/ANSI Version 4 Tape Support**

  Provides support for ISO/ANSI Version 4 tapes (ISO 1001–1986(E) and ANSI X3.27–1987), making data interchange with non-S/390 platforms possible. Unlike Version 3, data can be in any code page, but the tape labels must be in seven-bit ASCII, as was the case with Version 3. Data conversion is supported on read and write access for these tapes for a wide range of code pages. Data conversion can also be suppressed if desired. The 2K blocksize limit for data has also been removed.

- **NaviQuest Enhancements**

  Enhancements to the NaviQuest function of DFSMSdfp allow users to

  - Define, Alter, or Display
    - data class
    - storage class
    - management class
    - aggregate group
    - SMS base configuration
  - Define, Alter
    - VIO, Dummy, Object, or Object Backup

- **Virtual Tape Server (VTS) Import/Export Support**

  DFSMS/MVS supports the importing and exporting of VTS tape data removing the "closed store" limitation of the VTS. OAM enhances its support for the Virtual Tape Server (VTS) by supporting the physical removal of logical volumes from a VTS (export) and their reentry into a VTS (import). OAM provides new commands and programming interfaces and enhancements to existing installation exits to support these VTS capabilities. This support will be available as SPEs on DFSMS/MVS 1.4.0 and DFSMS/MVS 1.5.0.

  SMS support includes a pre-ACS interface that can be used by tape management systems to set new read-only variables.

  DFSMSrmm support includes recording that volumes within a VTS are "logical" and tracking the relationship between exported logical volumes and their "container" stacked volume.

- **Sysplex Catalog Alias**

  Symbolic substitution of alias names for user catalogs and non-VSAM data sets is now possible.

  The symbolic substitution allows a different user catalog name to be resolved on each system. For example, it allows an alias name of IMSCAT to resolve to IMS.PRODCAT on System A and IMS.TESTCAT on System B. Symbolic substitution for user catalog names is done at IPL, catalog address space restart, or when the master catalog is updated by another shared system.

  For non-VSAM data sets, a symbolic substitution allows one to use a single alias name to refer to different data set names on different systems. For example, an alias name of SYS1.PL1LIB can resolve to SYS1.R4PL1LIB on System A and SYS1.R5PL1LIB on System B. Symbolic substitution for non-VSAM data sets is done at time of use.

DFSMSrmm Enhancements

Enhancements to DFSMSrmm have been made in the following areas:

- Support for non-shared ICF catalogs in a shared DFSMSrmm environment
- Improved inventory management through dynamic tracking of changes in catalog status of tape data sets
- Support for a special DD statement that contains special handling information needed by tape operators for better disposition control
- Support for enhanced retention and vaulting policies by allowing separate definitions for retention and vaulting
- Easier access to information in the DFSMSrmm control data sets using an application programming interface (API)
- Support for obtaining information for multi-volume and multi-dataset chains using a single TSO subcommand or API request
- Support for creating ISO/ANSI Version 4 labels
- Support for recording information about ISO/ANSI Version 4 tape labels and for additional processing during OPEN processing when creating ISO/ANSI labels on a tape volume
- Support for changing volume serial number or data set name independently of catalogs and tape labels using enhanced DFSMSrmm subcommands

DFSMShsm Enhancements

Enhancements to DFSMShsm have been made in the following areas:

- Reduced overhead for incremental backup has been achieved by overlapping input/output operations with catalog and HSM control data set update operations
- Removal of the 4 gigabyte restriction that applied to each DFSMShsm control data set cluster by allowing the DFSMShsm control data sets to be allocated as VSAM EA data sets when the CDSs are accessed in RLS mode
- Serialization of resources at the HSMplex level for multiple HSMplexes that may exist in a single GRSplex, eliminating contention between HSMplexes for resources. Previously, if an installation had more than one HSMplex in a sysplex, each HSMplex would block the others when updating the DFSMShsm control data sets in non-RLS mode or when performing such functions as L1 to L2 migration.
- Enabling "secondary hosts" to take over "primary host" functions when a primary host failure occurs. Additionally, enabled hosts can take over Secondary Space management functions.
- Support has been added for the new VSAM functions of:
  - Partial release for data sets other than KSDSs
  - Extended addressability for all non-KSDS data sets
- Discontinuance of reel-type tapes for migration and backup output.

DFSMSdss Enhancements

Enhancements to DFSMSdss have been made in the following areas:

- Support for Extended Format VSAM data sets, including data set COPY, logical data set DUMP and RESTORE, and logical RELEASE support.
- Filtering by STORGRP, which allows the specification of a storage group name, rather than the volume serial numbers of all the volumes in the storage group, during data set COPY, logical DUMP, and logical RELEASE.

Object Access Method (OAM) Enhancements

OAM has been improved in the following ways:

- Any OAM object may be accessed from any instance of OAM that belongs to the same OAM XCF group in a parallel sysplex regardless of which OAM originally stored the object or which media type (DASD, 3995 optical, or tape) the object resides on.
- Removes the restriction of the 100–object storage group limit and associated restriction on the high-level qualifier used for each object storage group. The customer now has the capability of defining more or less than 100 object storage groups depending on the requirements of the installation. The high-level qualifiers can be user defined.
- The concept of pseudo libraries is changed to allow a customer to define pseudo libraries as a collection of operator-accessible optical drives and shelf-resident optical volumes not necessarily of the same device and media type as was a criteria in previous releases. Also, there is no restriction on the number of pseudo libraries that can be defined in an active SMS configuration.
- OAM provides support for an additional optical drive, 3995-SW4, that supports a media with 8 times the capacity of single-density WORM or rewritable optical disk media. This drive can read data from double-density WORM, CCW, or rewritable media. It can read from and write to quad- and 8x-density WORM, CCW, or rewritable optical media. This drive can be used in any of the 3995-Cxx optical libraries; however, a real optical library cannot contain a mix of quad- and 8x-density drives inside the same physical library.
- There is a new OAM RELABEL command that allows you to rename a volume serial number for a previously defined 3995 volume.
- There is new support for reformatting a 3995 optical disk cartridge using the OAMUTIL command. Reformatting these optical disk cartridges increases the usage of optical resources within the optical environment.

Chapter 3. An Overview of Storage Management With DFSMS/MVS

In a storage environment managed by the Storage Management Subsystem (SMS), your enterprise establishes centralized *policies* for how your hardware resources should be used. These policies balance your available resources with your users' requirements for data availability, performance, space, and security. SMS implements these policies and handles most of your storage management tasks. This frees users from manual storage administration and makes more efficient use of your storage resources.

This chapter briefly discusses how DFSMS/MVS can help you manage your storage. It is intended as a high-level overview. For a more detailed discussion of how each of the DFSMS/MVS functional components can help you manage storage, see the appropriate chapters in this book, listed below.

- For more information about DFSMSdfp, see:
  - Chapter 4, “Storage Management with DFSMSdfp” on page 23
  - Chapter 5, “Data Management with DFSMSdfp” on page 41
  - Chapter 6, “Program Management with DFSMSdfp” on page 61
  - Chapter 7, “Device Management with DFSMSdfp” on page 69
- For more information about DFSMSdss, see:
  - Chapter 9, “Role of the Functional Component DFSMSdss” on page 83
  - Chapter 10, “DFSMSdss Facilities” on page 91
- For more information about DFSMShsm, see:
  - Chapter 11, “Role of the Functional Component DFSMShsm” on page 95
  - Chapter 12, “DFSMShsm Facilities” on page 117
- For more information about DFSMSrmm, see Chapter 13, “Role of the Functional Component DFSMSrmm” on page 123

Implementing Your Storage Management Policies

The policies defined by your installation represent decisions about your resources, such as:

- What performance objectives are required by data?
- When and how to back up data?
- Whether data sets should be kept available for use during backup or copy?
- How to manage backup copies kept for disaster recovery?
- What to do with data that is obsolete or seldom used?

To implement a policy for managing storage, your storage administrator defines classes of space management, performance, and availability requirements for data sets at your installation. For example, the administrator can define one storage class for data entities requiring high performance and another for those requiring standard performance. Then, the administrator writes Automatic Class Selection (ACS) routines that use naming conventions, or other criteria of your choice, to automatically assign the classes that have been defined to data as that data is

© Copyright IBM Corp. 1980, 1999

created. These ACS routines can then be validated and tested. When the ACS routines are started and the classes (also referred to as constructs) are assigned to the data, SMS uses the policies defined in the classes and applies them to the data for the life of the data. Additionally, devices with various characteristics can be pooled together into storage groups so that new data can be automatically placed on devices that best meet the needs for the data.

DFSMS/MVS facilitates all of these tasks by providing menu-driven, fill-in-the-blank panels with the Interactive Storage Management Facility (ISMF). ISMF panels make it easy to define classes, test and validate ACS routines, and perform other tasks to analyze and manage your storage. Note that many of these functions are available in batch through NaviQuest.

Managing Data Placement with the Storage Management Subsystem

SMS simplifies the allocation process in the following ways:

Use Templates to Simplify Allocation of Data Sets

The storage administrator provides standard models or templates, each with a set of commonly used logical data set attributes, that determine the data set's processing characteristics. This allows users to reference these templates when creating data sets instead of having to continually respecify frequently needed attributes in JCL.

Place Data on Appropriate Storage Devices

DFSMS/MVS can select the type of storage device most appropriate for your data based on criteria you set. For example, if you are creating an image object, which is a named stream of bytes that has no internal orientation, DFSMS/MVS can place the data on an appropriate disk storage volume, also called a direct access storage device (DASD). If you are defining a data set, DFSMS/MVS can select an appropriate tape or DASD volume to hold the data.

Balance Use of DASD Volumes

You can balance the use of your DASD volumes and use them more efficiently by allowing DFSMS/MVS to manage data placement. For example, DFSMS/MVS can help you avoid situations in which one user group runs out of space and wastes system resources by repeatedly migrating and recalling relatively active data, while another user group has excess space that is either unused or used for data that is rarely needed.

Optimize Use of Tape Media

DFSMS/MVS can help you optimize your use of tape media. Reports provide you with the information you need to effectively implement IBM's recommended tape mount management methodology. Using tape mount management techniques, you can write your ACS routines to redirect your tape allocations to DASD, then use the DFSMShsm functional component to automatically move the data sets to tape as a group. You use all of your tape cartridge's capacity because the system places as many data sets on a single cartridge as possible. Compacting your data on tape helps you fit more data on each cartridge, further reducing the number of needed tape mounts.

With the pre-ACS interface available via SPE, tape management systems can assist in determining when tape allocation requests are not candidates for DASD by identifying those tape data sets which require movement to another location.

Another method of improving the utilization of your tape media is with the usage of the IBM Magstar 3494 Virtual Tape Server (VTS).

Create Libraries of Tape Cartridges Associated with Tape Devices

With DFSMS/MVS, you can manage your tape drives more efficiently by creating *storage groups* of tape cartridges that are associated with one or more pools of tape drives. By creating tape libraries, or named collections of storage groups and tape devices, you can balance tape use across tape drives and distribute the work for operators more evenly. Associating tape cartridges with tape drives also helps you ensure that tapes are mounted on adjacent devices.

Create Libraries of Optical Devices

You can manage your operator-accessible optical drives more efficiently by grouping them and defining optical libraries using ISMF. The definitions are then used in conjunction with other SMS constructs (storage class, management class, and storage group), SMS ACS routines, and the Object Access Method (OAM), a part of the DFSMSdfp functional component, to manage the object data. You can then use your definitions of automated optical libraries with your definitions of pseudo optical libraries to more efficiently manage the placement and migration of your object data.

Automate Your Tape and Optical Operations

You can increase the efficiency of your tape and optical operations by automating your tape and optical libraries. An installation using DFSMS/MVS can have automated and non-automated tape and optical libraries. In a tape library automated by an IBM 3494 or 3495 Tape Library Dataserver or in an automated optical library, the tasks of retrieving, storing, and controlling the mountable volume are performed automatically, allowing volumes to be mounted and removed without human intervention.

DFSMS/MVS can help you manage the movement and expiration of optical disks. With OAM, re-writable optical disks in automated optical libraries can be automatically reused when all the data has expired. When all the data expires on optical disks that are not re-writable, the disks can be removed from the automated optical library.

Virtual Tape Servers, Devices, and Volumes

You can reduce the number of physical cartridges, devices, and automated tape libraries that are needed to store data by implementing a IBM Magstar 3494 Virtual Tape Server (VTS). This virtual tape subsystem consists of virtual tape devices, virtual tape volumes, tape volume cache (DASD), and hierarchical storage management software.

Tape read and write commands are used for all I/O between the host and the virtual tape subsystem. When data is written to the VTS it is buffered in tape volume cache. The hierarchical storage management software will later write data to the tape media so that it is fully utilized. Because the data is buffered, writing to tapes is faster as it is not delayed by tape mounts. For example, mounting of scratch tapes is instantaneous when using a VTS.

Import/Export

DFSMS/MVS supports the importing and exporting of VTS tape data. For example, VTS tape data can be exported (physically ejected) from a virtual library for routine storage management tasks such as, application migration, data interchange, off-site vaulting, exterior (non-library) shelf storage, or disaster recovery. Refer to *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for detailed information on the importing and exporting of VTS tape data.

Tape data on a logical volume can also be imported into a VTS library; or reintroduced into a VTS library. The latter operation is necessary for the reuse of volume serial numbers.

Manage Your Removable Media

You can use the DFSMSrmm functional component to manage your removable media at the shelf, volume, and data set levels. Shelf locations can be assigned for volumes that are kept available and for volumes that are stored at a vault for vital records or backups.

Tape operators can locate and mount tapes more easily because the drive display and operator mount message tell the tape operator the shelf location of the volume to be mounted. DFSMSrmm automatically records information about data sets on tape volumes so that you can manage the data sets and volumes more efficiently. When all the data sets on a volume have expired, the volume can be reclaimed and reused. Data that is to be retained can optionally be moved to another location.

No shelf information is maintained for VTS library logical volumes. When a logical volume is exported, DFSMSrmm records the volume serial number of the stacked volume as the “container” of that logical volume.

Managing Space

After a data set has been allocated, DFSMS/MVS continues to manage your storage to make the most efficient use of your resources. The following sections describe what you can do with DFSMSdfp and the DFSMShsm functional component.

Compress Data

DFSMSdfp supports data compression using the host hardware data compression for sequential and VSAM key sequenced data sets (KSDS). This lets you use your system buffers, I/O channels, and DASD more efficiently. You can also use the IBM RAMAC Virtual Array Subsystem to further compress this data, as well as all other data in the subsystem.

Migrate Low-Activity Data

If you have data on your DASD that you do not use very often, DFSMShsm can automatically move that data to other devices that might be slower but that also cost less per megabyte. For example, DFSMShsm can move your low-activity data sets onto slower DASD or tape volumes. DFSMShsm also help you save on storage costs by automatically compacting or compressing your low-activity data. You can use OAM to move your low-activity objects onto slower DASD or optical volumes.

Automatically Recall Migrated Data

Low-activity data that has been moved remains easy to access. DFSMShsm automatically recalls it when you need it, using the DFSMSdss functional component, the same efficient data mover that moved the data from its original volume. Whenever DFSMSdss moves a data set to a new device, it automatically converts the data if the different device type requires it. You can also use DFSMSdss to move data between devices of different types or between system-managed volumes and non-system-managed volumes, and DFSMSdss automatically converts the data if required.

Delete Expired Data Sets

You can assign a management class to SMS-managed data sets to indicate how long they should be retained: for a specific number of days since creation, a specific number of days not-used, or until a specific date is reached. DFSMShsm automatically deletes eligible data sets during its space management cycle based on this management class specification. Using management class criteria instead of a specific expiration date or retention period lets you change the management class more easily and apply the change to all the data sets using that management class. By using management class criteria in this way, whenever the data set is accessed for output or update, you avoid an operator message requesting update authorization.

Reclaim Wasted Space on DASD

DFSMShsm automatically deletes expired or temporary data on your DASD volumes and lets you reuse that space for new data. DASD volumes holding more active data or lower-activity data are managed in this manner. DFSMSdss eliminates or reduces the fragmentation of free space.

| Your installation can choose to have DFSMShsm automatically reclaim your excess
| requested space once a day or choose to have DFSMSdfp reclaim excess
| requested space each time a data set is closed.

Reclaim Wasted Space on DFSMShsm-owned Tape Volumes

When your DFSMShsm-owned tapes no longer contain a certain percentage of valid data, you can have DFSMShsm transfer the valid data from multiple tapes onto fewer tapes. Consolidating the data in this manner allows you to minimize the amount of space on tape holding obsolete data.

DFSMS/MVS also identifies removable media volumes that you can reuse directly because all the data on those volumes has expired. The DFSMSrmm functional component determines when a cartridge in an automated tape library holds only expired data, so that the volume can automatically be reused for new data.

Managing Availability

DFSMS/MVS helps ensure that your data remains available by automatically backing up your data according to the service levels you define. If your original data is accidentally lost, you can replace your originals with backup copies. With DFSMS/MVS, you can:

Automatically Back up Your DASD Data

DFSMShsm provides several automatic backup capabilities that work together. By having DFSMShsm automatically back up any new or changed data, you can keep your backups current but reduce how often you have DFSMShsm back up entire DASD volumes. When you want to recover a data set or a volume, DFSMShsm uses all sets of backups to provide you with the most current version.

Move Your Peer-to-Peer Remote Copy (PPRC) Data Easily with P/DAS

PPRC dynamic address switching (P/DAS) is a software function that provides the ability to redirect all application I/O from one PPRC volume to another PPRC volume with minimal application impact. P/DAS allows application-transparent switching of I/O to support the following tasks:

- Planned outages (device or subsystem)
- Device migration
- Workload movement

P/DAS commands allow the system operator to redirect application I/Os that are currently sent to the primary volume to go to the secondary volume of the PPRC pair instead.

The P/DAS function is available on MVS/ESA 5.1 (with PTFs) and above in conjunction with all levels of DFSMSdfp 1.2.0 (with PTFs) and above. P/DAS operations are based upon the PPRC functions of the 3990 Model 6 Storage Control, and can be used in shared-DASD environments, including Parallel Sysplex environments. Refer to *DFSMS/MVS Remote Copy Administrator's Guide and Reference* for details on the P/DAS function.

Prepare for Disaster Recovery with Remote Copy

Remote copy offers two options for your disaster recovery and workload migration needs: extended remote copy (XRC) and peer-to-peer remote copy (PPRC). XRC makes use of the system data mover, and is a combined hardware and software solution. It provides an asynchronous copy operation of both system-managed and non-system-managed data to a second, remote location. XRC is a continuous copy operation, and is capable of operating over long distances (in conjunction with channel extenders). It runs unattended, without any involvement from the application users. In the unfortunate event of an unrecoverable error at your primary application site, the only data that is lost is the data that was in electronic transit between sites.

PPRC is the synchronous-copy disaster recovery solution. It is based on the 3990 storage control and is limited to ESCON distances. Refer to *DFSMS/MVS Remote Copy Administrator's Guide and Reference* for remote copy details.

Prepare for Disaster Recovery with Aggregate Groups

DFSMS/MVS provides you with another solution for dealing with unrecoverable errors through the aggregate group backup and recovery support (ABARS), which is a part of the DFSMShsm functional component. ABARS is a tool that lets you define those applications critical to your enterprise and back up each application as an entity. This entity is defined as an *aggregate group* of related data sets that require concurrent action. In the event of a disaster, you can use ABARS to restore critical applications after restoring the base operating systems. Identifying and backing up only critical data lets you reduce the total amount of data going offsite for disaster recovery.

You can define policies specifying how you want the aggregate copies to be managed, such as how many copies of versions you want, or how long you want to keep your extra versions, and then allow DFSMS/MVS to automatically implement those policies. You can use DFSMSrmm to manage the movement of both ABARS and DFSMShsm backup tapes. DFSMSrmm keeps track of where tapes are stored and can provide you with a list of tapes you need to move offsite for storage or to retrieve in case of an unrecoverable error. Note that automated tape libraries facilitate the storage of disaster backup tapes, since data can be written on tape devices in remote, unattended tape libraries. ABARS can also be used for application transfer from one site to another. Host systems can act as disaster recovery sites for each other when you use tape library partitioning support to set up sets of tape libraries at different locations accessed by the host systems.

Perform Backups Easily and Non-disruptively

DFSMShsm performs its backup and recovery functions efficiently and effectively. DFSMShsm uses DFSMSdss to quickly copy DASD data, automatically convert the data for the new DASD or tape volume if necessary, and optionally compact the backup copies to use less space. DFSMS/MVS can perform backup-while-open functions for all CICS VSAM File Control data sets. When combined with concurrent copy, this avoids discarding invalid backups and provides a consistent copy of the data for those data sets. Concurrent copy is very useful for KSDS and VRRDS data sets. With concurrent copy, the backup can be performed non-disruptively, as DFSMS/MVS addresses the need for continuous availability, by allowing you to make consistent backups or copies of DASD data while another program is using the data.

Access Data during Operating System or Hardware Outages

DFSMS/MVS can provide you with access to your data even during operating system or hardware outages. For example, you can restore data from tapes when the operating system is not functional. You can also maintain access to critical data sets on DASD in case of device problems by allowing DFSMS/MVS to maintain dual copies of your key DASD volumes.

Monitoring Your Policies

After you have established your installation's service levels and implemented policies based on those levels, you can use DFSMS/MVS facilities to see if your objectives have been met. Information on past use can help you develop more effective storage administration policies and manage growth effectively. Use the DFSMS/MVS Optimizer feature to help you monitor, analyze, and tune your policies.

Monitor DASD Use

DFSMS/MVS provides the access method services DCOLLECT facility to examine DASD volume usage, even if the volumes are not managed by the system, and to produce a data set containing the results. Looking at the past history and growth of your storage can help you decide when to add DASD volumes to accommodate future growth. DCOLLECT produces an audit trail of which users have modified various SMS constructs, extracts control information for data sets on DASD as well as pertinent data about migrated and backed-up data from DFSMShsm control data sets. This can then be used for billing or analysis.

Monitor Data Set Performance

DFSMS/MVS provides statistics in the system management facility (SMF) type 42 record that you can use to analyze data set performance. I/O statistics are available for both system-managed and non-system-managed data sets, providing information such as the total number of I/O operations and the average I/O response time. The DFSMS/MVS Optimizer feature uses these records to analyze and recommend changes to optimize performance of data sets and volumes. With Release 4, DFSMS/MVS provides a new data set performance metric, *I/O delay*, which provides for improved batch job analysis.

Additional statistics are also available for system-managed data sets residing on DASD volumes connected with 3990 Model 3 or 6 Storage Controls. These statistics include the total number of I/O operations for data sets that are candidates for caching and the number of I/O operations that actually used cache for those data sets. All of these statistics are also available by storage class.

Decide when to Consolidate Free Space on DASD

DFSMShsm provides a reporting facility to help you decide when you want to consolidate your data on DASD to eliminate or minimize free space fragmentation. You can print a report that gives you information about your DASD, including how much space is unused on each track, what percentage of space is unused, and how many tracks are empty.

Set Policies for DASD or Tape

DFSMS/MVS reporting facilities help you set policies for placing or migrating data onto DASD or tape. With the volume mount analyzer tool, you can print DFSMS/MVS reports on tape usage to obtain statistics about how often tapes are mounted or how often tape data sets are used. These reports can help you decide which classes of data you want to store onto DASD or tape and how long you want data to remain in active storage before being moved.

Use Reports to Manage Your Removable Media

DFSMSrmm reporting facilities help you manage your removable media. These reports help you keep track of where your tape volumes are and where they should be moved, by providing you with information about your volumes, shelves, and tape data sets. You can also audit use of your tape data by running security reports that list all occasions when a protected tape data set has been accessed.

Chapter 4. Storage Management with DFSMSdfp

The DFSMSdfp functional component provides the Storage Management Subsystem (SMS), which automatically assigns service requirements and attributes to new data when that data is created. SMS automatically controls system storage and decreases user concern about the physical characteristics of storage devices.

This chapter describes the attributes that can be assigned to data sets and objects through the data class, storage class, management class, and storage group constructs.

Using SMS Classes and Groups

Automatic class selection (ACS) routines assign classes to data, based on its requirements and attributes and select the target storage groups.

Data Class	Data definition parameters
Storage Class	Availability and accessibility requirements
Management Class	Data migration, backup and retention attributes
Storage Group	List of candidate allocation volumes

Figure 4 shows the SMS classes and groups that are assigned to data. The assigned classes and groups control the allocation, performance, and availability of:

- Data sets on DASD or tape volumes
- Objects on DASD, optical, or tape volumes

For a tape data set, the assigned classes and groups control the allocation, tape recording technique, and automation of the data set. Not every attribute in each class or group applies to every type of device, or to every type of data set.

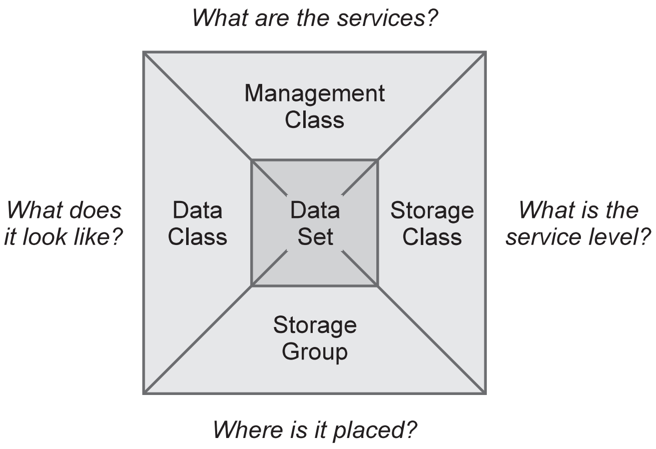


Figure 4. SMS and its Relationship to a Data Set

DFSMSdfp provides the storage administrator with Interactive Storage Management Facility (ISMF) menu-driven, fill-in-the-blank panels that the administrator can use to define classes of user needs. For example, the administrator might define one storage class for DASD data sets requiring high performance and another for DASD data sets requiring standard performance. The administrator might define

© Copyright IBM Corp. 1980, 1999

one storage class for objects requiring fast initial access response, which would place the objects on DASD, or another with lower initial access response, which would place objects on tape or optical media.

New data is automatically assigned to classes based on criteria the storage administrator selects, such as data naming conventions, and assigned to a storage group of devices that can meet the data's requirements. The values from the assigned classes are then used to determine the space management, performance, and availability requirements of the data for as long as the data exists.

In addition to these elements, the storage administrator defines *aggregate groups* of data sets that the DFSMShsm functional component of DFSMS/MVS backs up together to help you prepare for disaster recovery, application transfer, application archiving, or data migration among new business partners.

For further information about classes, groups, and ACS routines, see any or all of the following publications:

- *DFSMS/MVS DFSMSdfp Storage Administration Reference*
- *DFSMS/MVS Implementing System-Managed Storage*
- *MVS/ESA SML: Managing Data*
- *MVS/ESA SML: Managing Storage Groups*
- *DFSMS/MVS NaviQuest User's Guide*

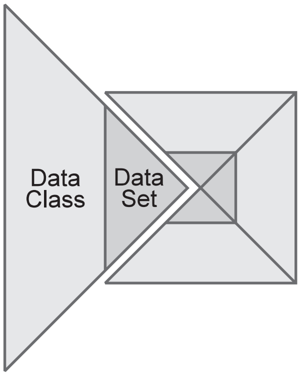
See Appendix A, “DFSMS/MVS Library and Related Publications” on page 129 for more information on these books.

Managing Data Set Attributes

Attributes, such as the expiration date or space requirements of a data set, can be managed effectively with data classes. A *data class* is a collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set. Most data class attributes apply to non-SMS-managed data sets.

Users need not specify data class attributes; an installation can have data class attributes assigned automatically based on the name of a data set. Any data attributes explicitly specified in JCL or equivalent allocation statements (for example, list or listing data sets generally have a low-level qualifier of LISTING) override those assigned through a data class ACS routine.

Figure 5 on page 25 shows the data set attributes that can be associated with a data class.



Record and Space Attributes
- Key Length and Offset
- Record Format
- Record Length
- Record Organization
- Space (Primary, Secondary, Avg Rec, Avg Value)

Volume and VSAM Attributes
- Compaction
- Control Interval Size
- Imbed
- Media Type and Recording Technology
- Percent Free Space
- Replicate
- Retention Period or Expiration Date
- Share Options (Cross Region, Cross System)
- Volume Count

Data Set Attributes
- Backup-While-Open
- Data Set Name Type
- Extended Addressability
- Extended Format
- Initial Load (Speed, Recovery)
- Log and Logstream ID
- Record Access Bias
- Reuse
- Space Constraint Relief and Reduce Space Up to %
- Spanned/Nonspanned

Figure 5. Data Class Attributes

A storage administrator can define data classes for use by jobs and users, providing a way to:

- Simplify JCL specification
- Use allocation defaults, specifying allocation values that make efficient use of storage
- Specify when a data set should be allocated in extended format
- Allocate VSAM data sets, using JCL, dynamic allocation, or the TSO ALLOCATE command
- Retry data set allocations on new volumes when allocation fails due to space constraints. Allocations are also retried when extending multivolume data sets to new volumes
- Let the system determine how many buffers and which buffering algorithms to use when a VSAM data set allocated in extended format is accessed using non-shared resources (NSR) and batch programs

For example, you can designate certain tape data sets for compaction to make more efficient use of the storage capacity of your tape cartridges. You can designate sequential data sets for sequential data striping, which distributes data for one data set concurrently across multiple SMS-managed DASD volumes, thereby improving I/O performance and reducing the batch window. You can also specify extended addressability for VSAM extended format data sets, to support data sets larger than 4 GB.

Managing Performance and Availability

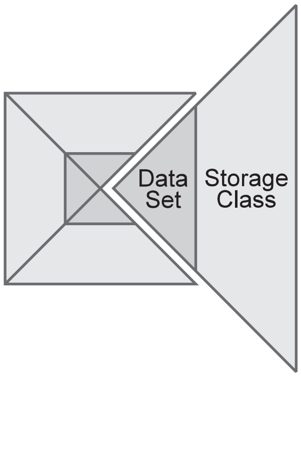
A *storage class* is a collection of performance goals and device availability requirements defined by the storage administrator. SMS uses it to select a device that can best meet those goals and requirements in terms of the performance of the device, the amount of space available on the volume, and how available a data set or object can be on that device.

The storage class availability and accessibility attributes are used with the 3990 Storage Control, and with the RAMAC Virtual Array, the RAMAC Scalable Array, and the RAMAC Electronic Array control units to provide a required level of service. When you define a storage class with a specific service level, SMS attempts to ensure that a data set associated with that storage class is allocated to the required hardware and that the caching abilities of the 3990 Storage Control are maximized to deliver the required service level. With the IBM 3990 Storage Control Model 3 or 6, when data is read or written, the storage class attributes can provide dynamic cache management at the data set level.

The storage class coupling facility (CF) caching attributes work with the CF to provide record-level sharing (RLS) for VSAM data sets in a Parallel Sysplex. When you define a storage class with a CF cache set name, data sets associated with that storage class are eligible for RLS. At data set open, if RLS is to be used for those data sets, they are cached in specific CF cache structures, or buffer pools, defined to SMS. Any CF weights defined for the storage class are considered at that time as well. CF weights are used to automatically determine the importance of customer data and the amount of storage in the cache structure to assign to the data. Storage classes with higher weight values are generally allocated more storage in the cache structure than storage classes with smaller weights, thereby improving performance.

Storage class is a required part of an active SMS-managed storage environment. Data sets on SMS-managed DASD and optical volumes with an assigned storage class are considered SMS-managed.

Figure 6 shows the attributes that can be associated with storage class.



Performance Objectives
- Direct Bias
- Direct Millisecond Response
- Initial Access Response
- Sequential Millisecond Response
- Sustained Data Rate

Availability Objectives
- Accessibility
- Availability
- Guaranteed Space
- Guaranteed Synchronous Write

Coupling Facility Caching Attributes
- Coupling Facility Cache Set Name
- Coupling Facility Direct Weight
- Coupling Facility Sequential Weight

Figure 6. Storage Class Attributes

By using the storage class attributes with ACS routines, your installation can:

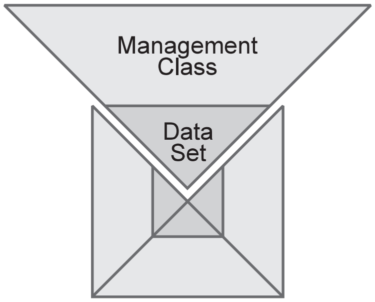
- Specify performance requirements at the data set or object level
- Use SMS to make optimal use of storage, matching performance and availability characteristics with user-specified needs
- Reduce user awareness of physical device characteristics
- Provide for the separation of physical device characteristics from a data set's logical requirements for performance, availability, and space
- Direct allocations to fault-tolerant devices, such as dual copy or RAMAC devices, to keep critical DASD data sets continuously available
- Use the concurrent copy function and DFSMSdss to keep DASD data sets almost continuously accessible while consistent backups or copies are made
- Use the sequential data striping function and request the sustained data rate that is required. DFSMS/MVS determines the number of devices to meet the requirement. Data for one data set is distributed and transferred concurrently for improved I/O performance.
- Designate tape data sets for placement in tape libraries
- Use record-level sharing processing for VSAM data sets that have been assigned to a coupling facility (CF) cache set
- Force allocation on one or more specific volume serial number

For more information on concurrent copy, see “Copying and Moving Data” on page 83. For more information on sequential data striping, see “Sequential Data Striping” on page 44. For more information on record-level sharing for VSAM data sets, see “Using VSAM Record-Level Sharing” on page 55.

Managing Space and Availability

A *management class* is a collection of management attributes defined by the storage administrator. For data sets, these attributes control retention, migration, backup, and release of allocated but unused space. For objects, the attributes control retention, backup, and class transition. You can also use management classes to define how your aggregate groups should be managed. DFSMShsm acts on many management class attributes.

Figure 7 on page 28 shows the attributes that can be defined in a management class.



Space Management Attributes
Partial Release

Expiration Attributes
Expire After Date/Days
Expire After Days/Non-Usage
Maximum Retention Period

Migration Attributes
Command or Auto Migrate
Level One Days Non-Usage
Primary Days Non-Usage

Generation Data Group Management Attributes
Number of GDG Elements on Primary
Rolled-Off GDS Action

Backup Attributes
Administrator or User Command Backup Versions
Auto Backup
Backup Frequency
Number of Backups (Data Set Deleted)
Number of Backups (Data Set Exists)
Retain Days Extra Backups
Retain Days Only Backup Versions
Backup Copy Technique

Class Transition Attributes
Object Class Transition Criteria

Aggregate Backup
Number of Versions
Retain Only Version
Retain Only Version Unit
Retain Extra Versions
Retain Extra Versions Unit
Copy Serialization
Abackup Copy Technique

Figure 7. Management Class Attributes

By using the management class attributes with ACS routines, your installation can:

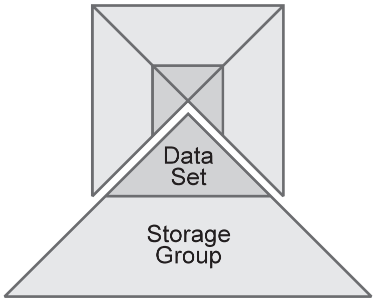
- Provide migration, backup, and deletion at the data set or object level. This enables the storage administrator to manage DASD, optical, and tape storage more effectively and to satisfy user requests for space management and backup.
- Specify whether a DASD data set must be copied concurrently using concurrent copy, should be copied concurrently if possible, or should be copied using normal backup processing techniques.
- Specify whether backup copies of an aggregate group of data sets must be backed up concurrently using concurrent copy, should be backed up concurrently if possible, or should be backed up using normal backup processing techniques.
- Reduce the effort users and the storage administrator require for storage management tasks.

See “Managing Space” on page 98 and “Managing Availability” on page 106 for more information on how DFSMShsm implements your management class policies.

Managing Data Placement

A *storage group* is a collection of storage volumes and attributes defined by the storage administrator. The storage administrator groups the volumes to meet a specific service or business strategy. Allocating data to storage groups can exploit the advantages of pooled volumes, improving your use of personnel and storage resources.

Storage groups are also used to specify which SMS-managed volumes should be processed by the DFSMShsm functional component of DFSMS/MVS and the systems on which they should be processed. Figure 8 shows the storage group attributes.



Pool Storage Group Attributes
Allocation/Migration High Threshold
Allocation/Migration Low Threshold
Auto Backup
Auto Dump
Auto Migrate
Backup System Name
Dump Classes
Dump System Name
Guaranteed Backup Frequency
Migrate System Name
Storage Group Status
Storage Management Subsystem Volume Status
Volume List

VIO Storage Group Attributes
Storage Group Status
VIO Maxsize
VIO Unit

Dummy Storage Group Attributes
Volume List

Object Storage Group Attributes
Library Names
Cycle Start Time
Cycle End Time
Drive Start Threshold
Mark Volume Full on First Write Failure
Volume Full Threshold

Object Backup Storage Group Attributes
Library Names
Drive Start Threshold
Mark Volume Full on First Write Failure
Volume Full Threshold

Tape Storage Group Attributes
Library Names

Figure 8. Storage Group Attributes

By using the storage group attributes with ACS routines, your installation can:

- Simplify creation of objects and segregate objects into groups according to customer usage and location within the storage hierarchy.
- Direct SMS to select volumes from storage groups for new data. For pool storage groups, the storage thresholds of each DASD volume are taken into account during the selection process so that space usage can be balanced across volumes within storage groups and DASD space can be used more efficiently.

For tape storage groups, the number of available scratch tape volumes in each library is also taken into account during the selection process, so that the storage resources can be balanced across storage groups.

- Simplify device installation and hardware configuration management. An administrator can add or delete devices through the ISMF Storage Group Application. An IPL is not necessary when making changes to storage groups.
- Simplify data set allocation for users. When SMS uses ACS routines to automatically direct new data sets to appropriate storage groups, users do not have to provide unit and volume information during allocation.
- Improve the balance of I/O activity across devices.

  When you allocate a new data set onto DASD, SMS selects an appropriate device to hold the data set, based on the performance requirements specified in the storage class for the data set, and based on the I/O activity of the devices in the storage group. This enables SMS to balance I/O activity across devices in the storage group. As a result, DASD performance becomes more consistent, reducing the need for manual tuning.

  There are several ways you can configure tape libraries to help you balance the usage of tape drives and minimize the time it takes to locate and transport tape volumes:

  – You can have an Automated Tape Library Dataserver (ATLDS) automate a tape library, mechanically locating, retrieving, mounting, removing, and storing tape cartridges.

  – You can use a manual tape library, allowing you to group together tape drives and tape volumes to create a library that can be system-managed from remote locations. This allows faster response times for mount and remove requests by tape operators, and provides more efficient use of tape storage facilities.

  – You can use stand-alone tape drives to handle requests against tape volumes outside of the tape library environment.
- Implement the tape mount management methodology, using ACS routines to redirect small tape allocations to DASD. DFSMShsm's automatic interval migration can move these redirected DASD data sets on a single tape cartridge and compacts them. This improves tape usage and significantly reduces tape mounts.

  When data set stacking is used within a single job step, the system tries to ensure that the stacked data sets are directed to the same media type (SMS-managed DASD, SMS-managed tape, or non-SMS-managed media) as that specified for the primary data set of the data set collection.
- Isolate application data for security or other business reasons.

Assigning Classes and Groups Using ACS Routines

You can use ACS routines to automatically determine the target storage group and assign data classes, storage classes, and management classes to SMS-managed data sets and objects. Data classes can be assigned to non-SMS-managed data sets as well. Automatic class selection provides centralized control over data set allocation on SMS-managed volumes. If SMS is activated, all new data set allocations are subject to automatic class selection. You can test ACS routines in batch through NaviQuest.

The storage administrator writes an ACS routine for each of the three types of classes and one to assign the storage groups. These routines, used together with the data class, storage class, management class, and storage group definitions, and the base configuration, define your site's SMS configuration. The storage administrator stores the information on this configuration in a source control data set (SCDS). See “Maintaining SMS Configurations” on page 33 for information on how you can activate or change these definitions.

Each time a new data set is allocated, SMS runs the ACS routines in the following order:

1. The data class routine is run first and determines whether to assign a data class to the data set.

   Note: Information obtained from a parmlib member is used to assign the data class for OAM objects being stored on tape volumes, and the ACS routines shouldn't interfere with this assignment.

2. The storage class routine is run next. If a storage class is assigned, the data set or object is put under SMS control.
3. Then the management class routine is run to assign a management class.
4. Finally, the storage group routine is run to determine candidate storage groups for the SMS-managed data set.

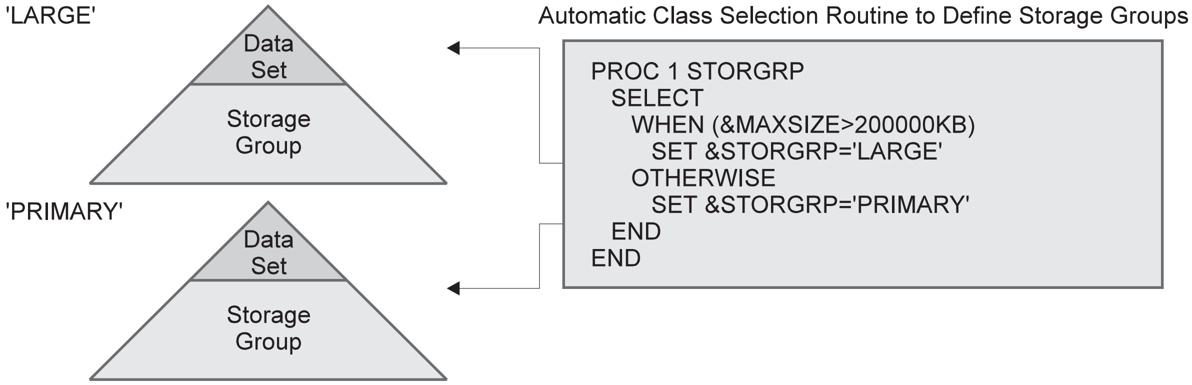
The selection of specific classes and groups is based on information from JCL or other allocation parameters. ACS routines can use parameters, such as data set name, volume serial number, job name, data set size, and others, to assign classes and groups to data sets.

If the pre-ACS interface is supported by tape management, additional information such as vault destination of a tape data set may be available.

If you are allocating a data set on a tape cartridge in a tape library, the system requests an available tape volume from a scratch pool and assigns it to the storage group you specify. In an automated tape library, the tape is automatically selected and mounted, then mechanically moved to a storage slot in the tape library. The next time that tape is needed, the tape is automatically located, retrieved, and mounted. The same tape volume allocation process takes place with the manual tape library; however, the library operator must manually retrieve and mount the tape on to the drive.

To implement the tape mount management methodology, write your storage class and storage group ACS routines so that they redirect tape allocations to a pool storage group. Since many tape allocations do not specify SPACE, you might also need to write your data class ACS routine to assign a data class providing space parameters to these data sets. DFSMShsm can automatically move the data sets together to tape at a later time. To make the data sets immediately eligible for migration to tape, define a management class that specifies 0 for the Level 1 Days Non-usage attribute.

Figure 9 on page 32 shows a portion of a sample storage group ACS routine used to assign very large data sets to the LARGE storage group and assign all other data sets to the PRIMARY storage group. Individual installations can determine their own standards for classifying data sets according to size. In Figure 9, data sets greater than 200 000 kilobytes (KB) are classified as large data sets; data sets smaller than those are placed in the primary storage group.



```
PROC 1 STORGRP
  SELECT
    WHEN (&MAXSIZE>200000KB)
      SET &STORGRP='LARGE'
    OTHERWISE
      SET &STORGRP='PRIMARY'
  END
END
```

Figure 9. Example of a Storage Group Automatic Class Selection Routine

Altering the Storage Class or Management Class of a Migrated Data Set

You can use the access method services ALTER command to alter a migrated data set's storage class, management class, or both, without having to recall the data set. This alter without recall function causes both DFSMShsm's and catalog's records to be updated and kept in synchronization.

This function is intended to replace the function previously supplied by DFSMShsm with the HALTER exec found in ARCTOOLS. The HALTER exec was rewritten to invoke the IDCAMS ALTER command.

Defining Aggregate Groups for Disaster Backup and Recovery

The storage administrator defines *aggregate groups* to enable the DFSMShsm functional component of DFSMS/MVS to back up and recover groups of data sets that require concurrent action. An aggregate group is a collection of data sets with identical backup attributes (such as type of storage medium, retention period, or destination) that have been pooled to meet a defined backup or recovery strategy. A storage administrator might group related data sets that:

- must be processed together as a group
- are required to run a user application
- can be used for disaster recovery
- are being transferred from one site to another

Use the aggregate backup and recovery support (ABARS) tool along with the aggregate groups you define to identify your enterprise's critical applications and back them up as an entity.

DFSMS/MVS lets you select a management class when you define an aggregate group with the Aggregate Group Application. Backup copies of that aggregate group are managed according to the aggregate group attributes you define for that management class. This lets you use the same management class attributes for multiple aggregate groups with the same management needs without having to define aggregate group attributes for each aggregate group individually.

See “Critical User Applications” on page 106 for more information on managing your aggregate groups with DFSMShsm.

Managing Objects

The Object Access Method (OAM) manages object data that can reside on DASD, optical, or tape volumes.

Objects are named streams of bytes. There are no restrictions on the data in an object because OAM does not recognize its content, format, or structure. For example, an object can be a scanned image of a document or an engineering drawing. Objects can be up to 50 MB in size. This size restriction applies to all supported devices.

The storage administrator defines optical libraries and optical drives to be used in processing objects within an optical storage environment, and relies on system allocated tape drives to handle objects to be written to tape volumes. Objects are assigned to object storage groups, and are directed to specific storage devices (DASD, optical, or tape) depending on their performance requirements dictated by an assigned storage class, data class, management class, and storage group.

You can use the SMF type 85 records created by OAM to analyze OAM activity during object support.

Maintaining SMS Configurations

SMS manages DASD, tape, and optical disk storage according to the information in the currently active *configuration*. It can also maintain information for the management of CF locking and data caching, enabling record-level sharing for VSAM data sets in a Parallel Sysplex environment.

An SMS configuration is composed of a set of data class, management class, storage class, storage group, tape library, and optical library and drive definitions, and the ACS routines that the storage administrator has defined to meet the needs of the site. Additionally, the SMS configuration includes the aggregate group definitions and the base configuration. The base configuration contains default information, such as default management class and default device geometry, and identifies the systems in the complex for which the subsystem manages storage.

The information that comprises an SMS configuration is contained in a source control data set (SCDS). You can define multiple SCDSs to describe different configurations. The configuration definition that is currently active is the current active control data set (ACDS). Any other SCDSs are inactive.

You can use ISMF to define, alter, and display the contents of an SCDS. You can also use ISMF to activate a new SMS configuration. Activating an SCDS causes its contents to be copied to the current ACDS without interrupting processing or disturbing inactive configuration definitions.

Defining System Group Name and 32-Name Support

Notes:

1. We strongly recommend that for any SMS system that is part of a Parallel Sysplex, all of the volumes in the SMS complex should be in the same Parallel Sysplex. Cross-system sharing functions, such as VSAM record-level sharing, PDSE sharing, RACF security and GRS serialization only work within the scope of a single Parallel Sysplex. They are not supported when the SMS complex extends beyond the Parallel Sysplex in which they are carried out.
2. We also recommend that you do *not* set up multiple SMS complexes sharing the same DASD. This not only requires extra work to maintain the duplicate SMS configurations but can also create problems such as running out of disk space, since one configuration cannot know about changes made to the other configuration, such as data set allocations and deletions, and storage group and volume status changes.

SMS system group name support allows you to specify either individual systems or an entire group of systems using a single name defined in the SMS configuration. The single name for a group of systems is the system group name. The system group name must match the Parallel Sysplex name defining the cooperative set of MVS systems. This set of MVS systems communicates and cooperates with each other using multisystem hardware components and software services.

With system group names, you might be able to combine SMS complexes into a single complex and eliminate additional control data sets.

SMS 32-name support allows you to specify up to 32 unique system names, system group names, or a combination of both in your SMS configuration. This lets you support the following:

- More than eight systems in a JES3 SMS complex, where system group names cannot be used
- System-level unique operations, such as VARY SMS commands, for more than eight unique systems, system groups, or a combination of unique systems and system groups
- More than eight unique system variations with regard to connectivity, status and workload.

  For example, you could have more object and object backup storage groups and optical libraries, which cannot be shared between systems. You could also define different storage group and volume connectivity for systems handling on-line transaction processing, and for systems handling batch processing.

SMS can run in either 32-name mode or in *compatibility mode*. When your system runs in 32-name mode, the SMS configuration can only be shared with other DFSMS/MVS 1.3 or higher systems running in 32-name mode. When your system runs in compatibility mode, only eight system and/or system group names are supported in the SMS configuration. In compatibility mode, the configuration can be shared with systems running down-level releases of DFSMS/MVS or DFP, and with systems also running in compatibility mode. Once the control data sets are converted to 32-system mode, they cannot be reverted back to eight-system mode.

Using SMS in a Multisystem Environment

You can use system-managed storage in a multisystem environment in which the systems, or MVS images, share a common SMS configuration. SMS is the system focal point for initial and subsequent DASD space allocation. It balances allocation across the volumes in storage groups for users and for DFSMShsm and DFSMSdss functions, such as restore, recover, and recall.

Note: For information concerning optical library connectivity, refer to *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

Shared SMS control data sets contain a common set of classes and storage groups, ACS routines, and a base configuration that can be used consistently across the systems in a multisystem environment.

A multisystem environment requires the use of a cross-system serialization program to prevent the system from migrating data that is used on a system other than the one on which migration is performed. In the MVS environment, this function can be accomplished by activating global resource serialization (GRS).

Note: If you share data or catalogs with lower-level systems, you must apply program temporary fixes (PTFs) to prevent data contamination and catalog damage. See *DFSMS/MVS Program Directory* for details on which PTFs are required.

System-Managed Storage Coexistence

SMS-managed data sets and volumes can coexist with non-SMS-managed data sets and volumes, which simplifies the migration to SMS-managed storage.

You can share source modules stored in PDSEs on SMS-managed volumes among multiple processors in a complex if DFSMS/MVS or MVS/DFP Version 3 Release 3 or higher is installed on each system. You can share program objects stored in PDSEs on SMS-managed volumes among multiple processors in a complex if DFSMS/MVS is installed on each system. You can share any member of a PDSE on SMS-managed volumes among multiple processors in a complex if DFSMS/MVS is installed on each system.

See *DFSMS/MVS Implementing System-Managed Storage* for details about SMS-managed and non-SMS-managed data set coexistence.

Defining Use of the Coupling Facility for VSAM Record-Level Sharing

In order for DFSMSdfp to use the coupling facility (CF) for VSAM record-level sharing (RLS), you must define one or more CF cache structures to MVS and add them to the SMS base configuration. You must also define a CF lock structure.

The system selects a CF cache structure for a data set based on the policy set by the storage administrator. Cache structures are defined in the coupling facility resource manager (CFRM) policy, used by MVS to determine where to allocate cache structures. Up to eight cache structures can be associated with a single CF cache set, which is defined in the SMS base configuration. When a storage class associated with a data set contains a CF cache set name, the data set is eligible for record-level sharing and can be placed in a CF cache structure associated with the CF cache set. The system selects the best cache structure within the cache set defined for the storage class.

You can have up to 256 CF cache set definitions in the base configuration. Each cache set can have up to eight cache structures defined to it, allowing data sets to be assigned to different cache structures in an effort to balance the load. The CF must have the same level of connectivity to systems using VSAM RLS as any storage groups used by those systems. This is to ensure that jobs running in the Parallel Sysplex have access to data in both the CF and the storage groups.

You must also define a single, non-volatile CF lock structure, IGWLOCK00, to be used to enforce VSAM RLS protocols and perform record-level locking. This lock structure must have global connectivity to all systems using VSAM RLS, so that it is accessible from all those systems.

See “Using VSAM Record-Level Sharing” on page 55 for more information on record-level sharing for VSAM data sets.

Migrating Data Sets to SMS-Managed Storage

ACS routines determine whether a new data set should be SMS-managed. The storage administrator defines ACS routines to direct selected data sets to SMS-managed volumes. With DFSMS/MVS, you can have your ACS routines direct your data sets to SMS-managed tape or DASD volumes.

As the migration to SMS-managed storage proceeds, the storage administrator modifies the routines to select more data sets to be SMS-managed. In this way, migrating to SMS-managed storage can occur gradually, minimizing user involvement and modifications to JCL or other allocation statements. You can also use DFSMSdss to migrate existing data sets to SMS-managed storage without data movement.

You can use NaviQuest to change and test ACS routines in batch. See *DFSMS/MVS NaviQuest User's Guide* for more information.

See *DFSMS/MVS Implementing System-Managed Storage* for details about data set migration.

Managing SMS Authorization

You can use RACF to control SMS resource protection at the user, group, and installation levels. RACF can:

- Authorize access to the SMS control data sets
- Authorize system managed storage maintenance functions
- Authorize use of management class and storage class, either through the RACF RESOWNER value or through the data set allocator
- Define class defaults in the user and group profiles
- Authorize access to the class fields found in the user and group profiles
- Define the data set owner in the data set profile
- Authorize access to ISMF functions

For more information about RACF, see “Resource Access Control Facility Protection” on page 57.

Interactive Storage Management Facility and SMS-Managed Storage

Interactive Storage Management Facility (ISMF) is a common interactive interface to DFSMS/MVS that helps you analyze and manage data and storage interactively. ISMF provides interactive access to the space management, backup, and recovery services of the DFSMShsm and DFSMSdss functional components of DFSMS/MVS, to the tape management services of the DFSMSrmm functional component, as well as to other products. DFSMS/MVS introduces the ability to use ISMF to define attributes of tape storage groups and libraries.

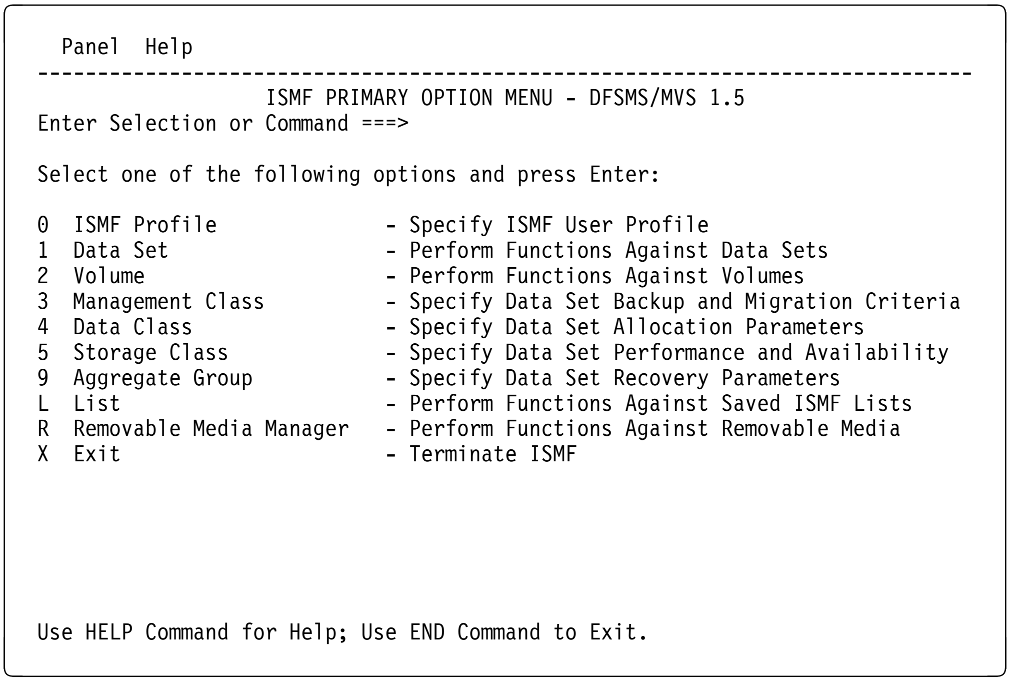
A storage administrator uses ISMF to define the installation's policy for managing storage by defining and managing SMS classes, groups, and ACS routines. ISMF then places the configuration in an SCDS. You can activate an SCDS through ISMF or an operator command.

Note: Information provided by the SETOAM command on the CBROAMxx parmlib member is used to establish and tailor object to tape support within an installation. This information can override or supplement information obtained from ISMF for an object being stored on to tape.

ISMF operates as an Interactive System Productivity Facility (ISPF) application. It is menu-driven with fast paths for many of its functions. ISMF uses the ISPF 4.2 data tag language (DTL) to give its functional panels on workstations the look of common user access (CUA) panels and a graphical user interface (GUI).

The ISMF Primary Option Menu for General Users

When you start a dialog, ISMF displays its primary option menu as shown in Figure 10, which shows the menu for general users:



```
  Panel  Help
------------------------------------------------------------------------------
                   ISMF PRIMARY OPTION MENU - DFSMS/MVS 1.5
Enter Selection or Command ===>

Select one of the following options and press Enter:

0  ISMF Profile             - Specify ISMF User Profile
1  Data Set                 - Perform Functions Against Data Sets
2  Volume                   - Perform Functions Against Volumes
3  Management Class         - Specify Data Set Backup and Migration Criteria
4  Data Class               - Specify Data Set Allocation Parameters
5  Storage Class            - Specify Data Set Performance and Availability
9  Aggregate Group          - Specify Data Set Recovery Parameters
L  List                     - Perform Functions Against Saved ISMF Lists
R  Removable Media Manager  - Perform Functions Against Removable Media
X  Exit                     - Terminate ISMF





Use HELP Command for Help; Use END Command to Exit.
```

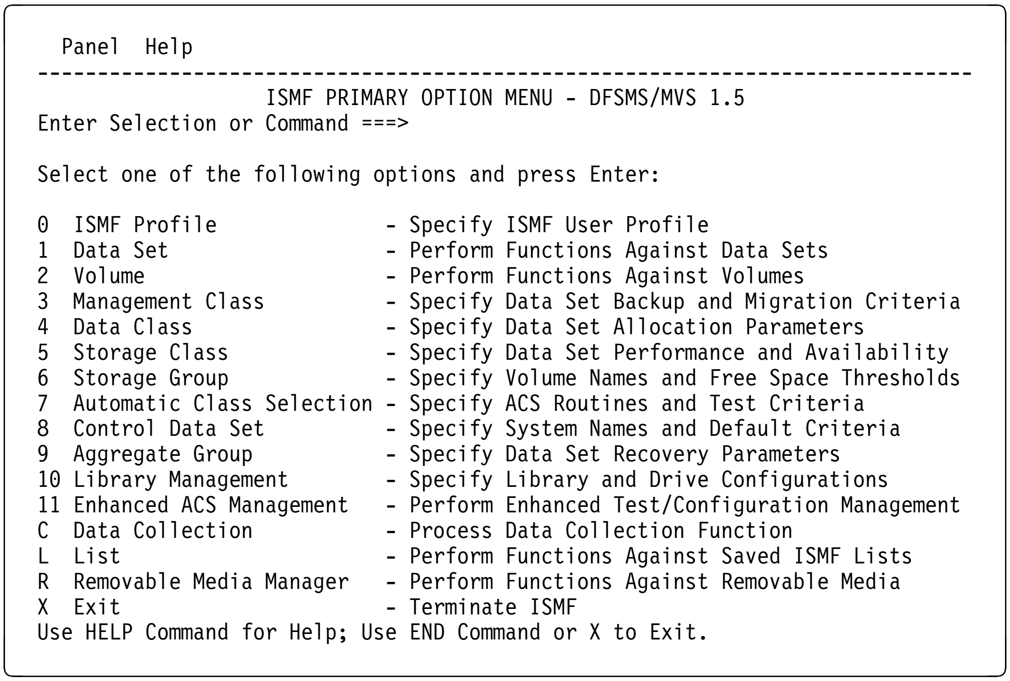
Figure 10. ISMF Primary Option Menu for General Users

You can perform functions against data sets, DASD volumes, tape volumes, and optical volumes. You can obtain information on the current definitions of SMS classes, or on data sets, volumes, and DFSMShsm storage. You can display and manage saved lists, print saved or generated lists, or directly reuse lists saved in the data set and volume applications. You can change profile data that controls such things as error logging or JOB statement information.

For additional information on SMS and ISMF for general users, see *DFSMS/MVS Using ISMF*.

The ISMF Primary Option Menu for Storage Administrators

The storage administrator uses ISMF to specify the definitions required to run SMS. Figure 11 shows the primary option menu for storage administrators.



```
  Panel  Help
------------------------------------------------------------------------------
                   ISMF PRIMARY OPTION MENU - DFSMS/MVS 1.5
Enter Selection or Command ===>

Select one of the following options and press Enter:

0  ISMF Profile             - Specify ISMF User Profile
1  Data Set                 - Perform Functions Against Data Sets
2  Volume                   - Perform Functions Against Volumes
3  Management Class         - Specify Data Set Backup and Migration Criteria
4  Data Class               - Specify Data Set Allocation Parameters
5  Storage Class            - Specify Data Set Performance and Availability
6  Storage Group            - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set         - Specify System Names and Default Criteria
9  Aggregate Group          - Specify Data Set Recovery Parameters
10 Library Management       - Specify Library and Drive Configurations
11 Enhanced ACS Management  - Perform Enhanced Test/Configuration Management
C  Data Collection          - Process Data Collection Function
L  List                     - Perform Functions Against Saved ISMF Lists
R  Removable Media Manager  - Perform Functions Against Removable Media
X  Exit                     - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 11. ISMF Primary Option Menu for Storage Administrators

Beginning with the primary option menu, ISMF provides you with dialogs for accomplishing many storage management tasks. You can define your installation's policies for managing storage by defining, altering, deleting, and copying SMS classes and groups and defining and managing ACS routines for DASD, optical, and tape storage environments. You can display online DASD volume space usage information, which you can use to change your SMS configuration so the system manages your storage most efficiently. You can define, alter, display, and list optical and tape libraries, optical drives, and the tape and optical volumes in the libraries. You can display and manage saved lists, print saved or generated lists, or directly reuse lists saved in the data set or volume applications.

Notes:

1. In addition to ISMF, you can use the DFSMS/MVS Optimizer Feature, a separately orderable feature of DFSMS/MVS, to analyze the proper settings for your SMS policies and functions.
2. ISMF only displays storage location information for tape volumes. If you want information about the data sets on tape volumes, use the DFSMSrmm ISPF dialog or the DFSMSrmm TSO subcommands. If you want information about the contents of tape volumes owned by OAM, use SQL Processing Using File Input (SPUFI).

For additional information on SMS and ISMF for storage administrators, see *DFSMS/MVS DFSMSdfp Storage Administration Reference*.

Using the NaviQuest Tool for Batch

You can access the NaviQuest tool by using option 11 (Enhanced ACS Management) from the ISMF Primary Option Menu for storage administrators. Use NaviQuest to migrate to SMS and maintain your SMS configuration. NaviQuest also offers code and tools to assist in SMS testing, implementation, and reporting.

With NaviQuest you can:

- Automatically create test cases
- Run ISMF in batch mode using REXX EXECs
- Create reports using ISMF saved lists
- Create reports with DCOLLECT and VMA data
- Use ACS code fragments as models when creating your own ACS routines
- Create or modify management classes
- | Create, modify, or display SMS configurations
- Add or delete volumes to storage groups

NaviQuest was developed specifically to be used with the DFSMS Fast Implementation Technique (FIT) for system-managed storage. NaviQuest Release 1 Version 1, a standalone product, supports the following releases: MVS/DFP 3.3, DFSMS/MVS 1.1 and 1.2. As of the DFSMS/MVS 1.3 release, NaviQuest is part of DFSMSdfp.

See *DFSMS/MVS NaviQuest User's Guide* and *DFSMS/MVS Implementing System-Managed Storage* for more information. See *Get DFSMS FIT: Fast Implementation Techniques* for information on using DFSMS FIT.

Chapter 5. Data Management with DFSMSdfp

DFSMSdfp data management elements handle the organization and storage of data in an environment with large real and virtual storage capabilities. These elements perform the tasks of organizing and processing the data in a data set or object, managing the placement of data sets or objects, and controlling access to data.

This chapter describes how DFSMSdfp data management elements are used to create and maintain data sets, manage data sets and volumes, and control access to data.

Creating and Maintaining Data Sets

DFSMSdfp provides several access methods for formatting and accessing data. An *access method* defines the organization of the data in a data set and the technique by which the data is stored and retrieved. DFSMSdfp access methods have their own data set structures to organize data, macro instructions to process data sets, and utility programs to manipulate data sets.

Do not confuse the DFSMSdfp access methods with the access methods of other MVS products, such as the Information Management System (IMS) access methods HIDAM, SHIDAM, GSAM, and OSAM.

DFSMSdfp Access Methods and Associated Data Set Structures

Various access methods create different types of data entities and use different techniques for accessing and storing data. In addition to the access methods described here, DFSMS/MVS supports the basic direct access method (BDAM) and the indexed sequential access method (ISAM) for compatibility with previous operating systems. All DFSMSdfp access methods, except ISAM, support 31-bit addressing for most or all macros and allow data buffers to reside above the 16 MB line.

Object Access Method (OAM)

OAM processes named byte streams (objects) that have no record boundary or other internal orientation that the system maintains. These objects are recorded either on DASD in a DB2 database, or on an optical or tape storage volume. The maximum object size is 50 MB regardless of which type of media is used.

Virtual Storage Access Method (VSAM)

VSAM arranges records by an index key, by relative byte address, or by relative record number. Data organized by VSAM is cataloged for easy retrieval and is stored in one of five types of data sets.

For more efficient use of virtual storage, buffer pools can be shared among all data sets except linear data sets, using globally or locally shared buffer pools. When you use the local shared resources facility to obtain VSAM buffers, you can specify VSAM hiperspace, which provides a high performance method for accessing expanded storage. When you specify VSAM hiperspace, VSAM buffers are located in expanded storage to improve the processing of VSAM data sets. VSAM lets you create buffers, user exits, shared resource pools, and some control blocks in virtual storage above the 16 MB line.

© Copyright IBM Corp. 1980, 1999

DFSMS/MVS also supports record-level data sharing for VSAM data sets in a Parallel Sysplex, using the locking and caching features of the coupling facility. See "Using VSAM Record-Level Sharing" on page 55 for more information.

The types of VSAM data sets are:

Key-sequenced data set (KSDS) contains records in order by a key field and can be accessed by the key or by a relative byte address. If they are allocated in extended format, these data sets can also contain compressed data and can be larger than 4 GB.

Any VSAM key-sequenced data set allocated in extended format must be cataloged in an integrated catalog facility (ICF) catalog. All other VSAM data sets can be cataloged in an integrated catalog facility or VSAM catalog.

Entry-sequenced data set (ESDS) contains records in the order in which they were entered and can only be accessed by relative byte address.

Relative-record data set (RRDS) contains records in order by relative-record number and can only be accessed by this number. Relative records can be fixed-length or variable-length.

Variable-length relative-record data set (VRRDS) contains records in order by relative-record number and can only be accessed by this number. These data sets let you access relative records that have a variable length.

Linear data set (LDS) contains data that can be accessed as byte-addressable strings in virtual storage; it contains none of the control information that other VSAM data sets hold. Linear data sets must be cataloged in an integrated catalog facility catalog. Linear data sets can also be accessed using the MVS data in virtual (DIV) access method. VSAM record-level sharing (RLS) is not available for VSAM linear data sets.

You can use the VSAM interface to access data in OS/390 UNIX files. With VSAM, the OS/390 UNIX file is accessed as an ESDS.

Basic Sequential Access Method (BSAM)

BSAM arranges records sequentially in the order in which they are entered. Records are stored in physical blocks and retrieved as requested. This is called *basic* access. You can use BSAM with physical sequential data sets or members of partitioned data sets or partitioned data sets extended (PDSE). BSAM data sets can be compressed and striped.

| You can also use the BSAM interface to access data in OS/390 UNIX files. With BSAM, the OS/390 UNIX file is accessed as if it were a single-volume, physical sequential data set residing on DASD.

Queued Sequential Access Method (QSAM)

QSAM arranges records sequentially in the order in which they are entered. QSAM collects records into blocks. This is called *blocking*. QSAM anticipates the need for records based on their order, and, to improve performance, reads these records into storage before they are requested. This is called *queued* access. You can use QSAM with physical sequential data sets, or with members of partitioned data sets (PDSs) or partitioned data set extended (PDSEs). QSAM data sets can be compressed and striped.

| You can use the QSAM interface to access data in OS/390 UNIX . With QSAM, the OS/390 UNIX file is accessed as if it were a single-volume, physical sequential data set residing on DASD.

Basic Partitioned Access Method (BPAM)

BPAM arranges records as members of partitioned data sets or PDSEs. A partitioned data set or PDSE includes a directory that relates member names to locations on the DASD volume, in order to retrieve individual members.

Hierarchical File System (HFS)

DFSMS/MVS provides access to enterprise data in an open system environment. You can use the standard BSAM, QSAM, and VSAM access methods to access data in HFS files. The HFS data set contains the HFS file structure. This structure is a framework of directories and HFS files called a file system. The structure resembles a tree with subtrees, each consisting of a directory and all its related files. The HFS files are identified and accessed by specifying the *path* leading to them.

With DFSMS/MVS, OS/390 UNIX manages HFS files, providing access to the data they contain. You can use DFSMShsm and DFSMSdss to backup and recover the whole HFS data set. In a sysplex environment the file system must be backed up on the system that has done the mount. You can use the ADSM V2 OS/390 UNIX client or the ADSM V3.1 OS/390 UNIX client to provide backup and recovery for the individual HFS files within an HFS file system.

For BSAM and QSAM, the HFS file is accessed as if it were a single-volume, physical sequential data set residing on DASD. For VSAM, the HFS file is accessed as an ESDS. Since HFS files are not actually stored as physical sequential data sets or ESDSs, some processing restrictions might apply, and certain macros and services might have incompatibilities when HFS files are processed.

For more information, see:

- *DFSMS/MVS Using Data Sets*
- *DFSMS/MVS DFSMSdfp Advanced Services*
- *DFSMS/MVS Macro Instructions for Data Sets*
- *OS/390 UNIX System Services User's Guide*

Extended Format Data Sets

Both SAM sequential data sets and VSAM data sets can be allocated in extended format. An extended-format data set is allocated on system-managed DASD attached to a controller supporting Extended Platform.

The following are possible for data sets allocated in extended format:

- Compressing both sequential data sets and VSAM key-sequenced data sets
- Sequential data striping (BSAM and QSAM only)
- Selecting whether to use the primary or secondary space amount when extending VSAM extended-format data sets to multiple volumes
- Supporting VSAM data sets larger than 4 GB
- Using system-managed buffering for VSAM batch programs that access the data sets in NSR mode
- Supports partial release for all VSAM data set organizations

Compressing Extended-Format Data Sets

Compression is a technique used to reduce the space required to store data as well as improve buffering and caching, reduce channel utilization, and reduce I/O rate. The compression feature exploits the processor compression hardware if it is installed. If the hardware is not available, a programming equivalent is used. In many cases, when the programming equivalent is used, the benefits can affect processor time.

Both SAM and VSAM support a compression technique which uses Dictionary Building Blocks (DBBs) to compress data. In DFSMS/MVS 1.4, you can request compression for SAM data sets that is specifically tailored to the data set, using the COMPRESS(TAILORED) parameter in the IGDSMSxx parmlib member or the COMPACTION attribute in the data class construct. Note that once they have been compressed using tailored compression, these data sets cannot be accessed from lower-level systems.

To be eligible for compression, the data set must be a system-managed data set allocated as an extended-format data set. System-managed sequential data sets accessed through the BSAM and QSAM access methods can be compressed; so can the data component of a VSAM key-sequenced data set. IMS data sets are not eligible for compression.

You request compression through the data class COMPACTION attribute. See *DFSMS/MVS Using Data Sets* for more information.

Sequential Data Striping

Sequential data striping gives you the option of distributing data for one data set across multiple system-managed DASD volumes. By transferring data concurrently to or from multiple devices, I/O performance can be improved and the batch window can be reduced. Sequential data sets accessed through the BSAM and QSAM access methods are eligible for sequential data striping.

See “Using SMS Classes and Groups” on page 23 for more information on data class and storage class definitions that support sequential data striping.

Selecting Space Allocations on New Volumes

In releases prior to DFSMS/MVS Version 1 Release 2, if VSAM data sets extend beyond the primary volume, the initial space allocation on the new volume is the primary amount. When non-VSAM data sets extend beyond the primary volume, the initial space allocation on the new volume is the secondary amount.

You can use the data class ADDITIONAL VOLUME AMOUNT attribute for VSAM extended-format data sets to select the amount of space to be used when extending to multiple volumes. Based on the value you specify, DFSMS/MVS uses either the primary or secondary volume amount when extending the data set to multiple volumes.

Extended Addressability

To allow VSAM data sets to contain more than 4 GB of data, DFSMS/MVS supports extended addressability for system-managed VSAM data sets allocated as extended format.

Sequential data sets that are not in extended format are limited in size to 65535 tracks per volume. Extended-format data sets can use all allocatable space or up to

| 123 extents per volume. With compression, an extended-format data set can hold even more data.

Notes:

1. You cannot access a VSAM key-sequenced data set allocated in extended format on any system prior to DFSMS/MVS 1.2. Additionally, you cannot open these data sets on DFSMS/MVS 1.2 if they have been allocated with the extended addressability option specified.
2. You cannot access extended-format RRDSs, ESDSs, VRRDSs, and LDSs on systems prior to DFSMS/MVS 1.5.
3. IMS data sets are not eligible for extended addressability or compression.
4. Other factors might limit the data set, such as 59 volumes for all components, 255 extents per component, 123 extents per component per volume.

Using System-Managed Buffering for VSAM Batch Programs

You can let the system optimize buffering algorithms and buffer size for extended format VSAM key-sequenced data sets accessed using non-shared resources on | DFSMS/MVS 1.4. You can let the system optimize buffering algorithms and buffer | size for extended-format VSAM data sets other than KSDSs on DFSMS/MVS 1.5. When you do this, you no longer have to use BatchLSR. The system selects buffering algorithms and buffer sizes based on the processing options that are specified when the data set is opened, and on the data set's storage class attributes. These can be overridden using the AMP parameter.

Data Management Macro Instructions

You can use *macro instructions* to create, maintain, and process all the data set types supported by the access methods described in “DFSMSdfp Access Methods and Associated Data Set Structures” on page 41. Macro instructions control data set allocation, input and output, and data security.

The DFSMSdfp data management macro instructions are described in:

DFSMS/MVS Macro Instructions for Data Sets
DFSMS/MVS DFSMSdfp Advanced Services

OAM OSREQ macro instructions are described in *DFSMS/MVS OAM Application Programmer's Reference* and the CBRXLCS macro information is available in *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Open/Close/End-of-Volume Macros

The open, close, and end-of-volume macros establish an environment where access methods can read data into buffers and return it to the correct location on DASD or tape.

The open function mounts the correct DASD or tape volume, prepares the volume for processing, and establishes a link between the system, the access method, and your program. When your program has finished processing the data set, the close function disconnects the link and removes the volume.

If an output data set on a DASD runs out of allocated space during processing, the end-of-volume function uses direct access device space management (DADSM) to allocate additional space. DADSM is described under “System Data Administration” on page 52.

For DASD, end-of-volume can automatically switch to the next volume of a multivolume data set and can verify any additional volumes to be mounted. When a system-managed data set is extended to another volume, end-of-volume selects another volume within the same storage group.

Checkpoint and Restart Functions

Checkpoint/Restart functions can be used to establish checkpoints during a program and to restart the job at a checkpoint or at the beginning of a job step.

A *checkpoint* is a designated point in the program at which information about the job is collected and recorded in a separate checkpoint data set. This information includes the contents of the program's virtual storage area and some related system control data. If the job fails, for example, or produces unusual output, the information in the checkpoint data set can be retrieved and the job restarted directly from this checkpoint.

MVS lets you restart a job automatically, with permission from the operator, or defer the restart until the job is resubmitted. In either case, you can restart from the checkpoint or from the beginning of the unsuccessful job step, avoiding the time-consuming process of rerunning the entire job from the beginning.

For more information about checkpoint/restart, see the following:

- *DFSMS/MVS Checkpoint/Restart*
- *DFSMS/MVS Using Data Sets*

Notes:

1. Checkpoint/Restart is not supported for extended-format data sets or PDSEs.

Data Management Utilities

You can use data management *utility programs* to perform a variety of tasks, such as moving or copying data.

- **Access Method Services**, also known as IDCAMS, creates and maintains VSAM data sets. With access method services, you can:
  - Define VSAM data sets
  - Define and build alternate indexes
  - Back up and restore VSAM data sets
  - Copy data sets
  - Print the contents of data sets
  - Delete data sets
  - Collect information about data sets
  - Examine the structural consistency of VSAM key-sequenced data sets
  - Control DASD cache
  - Diagnose catalog errors
  - Define system-managed libraries and volumes
  - Define extended addressability for an extended-format VSAM data set to support a data set size greater than 4 GB

- **IEBCOMPR** compares logical records in sequential or members of partitioned data sets, usually to verify the accuracy of backup copies.
- **IEBCOPY** copies and merges partitioned data sets and PDSEs and loads and unloads data to a sequential data set (tape or DASD). (The DFSMSdss DUMP and RESTORE commands can also be used to unload data to and reload data from a sequential data set.) IEBCOPY can also be used to alter RLD counts and to copy and reblock load modules. It will automatically convert load modules into program objects, or program objects into load modules, if the different formats of the input program library and output program library require this conversion. See "Using Utilities for Program Management" on page 65.
- **IEBDG** creates a pattern of test data for aid in debugging programs.
- **IEBEDIT** edits jobs and job steps into a single output data set.
- **IEBGENER** copies records from a sequential data set or converts sequential data sets into members of partitioned data sets. IEBGENER can also reblock and edit data sets. You can use DFSORT's ICEGENER utility as a more effective replacement for IEBGENER.
- **IEBPTPCH** punches or prints records from sequential or partitioned data sets.
- **IEBUPDTE** changes source language statements in sequential or partitioned data sets. IEBUPDTE can create and update program libraries and change data set organization between sequential and partitioned.
- **IEHINITT** labels tapes in stand-alone devices and automated tape libraries.
- **IEHLIST** lists entries in CVOLs, partitioned data set or PDSE directories, or volume tables of contents (VTOCs).

Besides these utilities, DFSMS/MVS also supports the IEBISAM, IEHMOVE, and IEHPROGM utilities for compatibility with previous operating systems.

Access method services commands are described in:

DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Access Method Services for VSAM

For more information on the data management utilities, see *DFSMS/MVS Utilities*

Character Data Representation and Data Conversion

Whenever you send textual (character) data, it is represented at the machine internal level by binary code—code which can vary among computer platforms and international languages. This can complicate how computer systems work together, especially in client/server scenarios where computer platforms are different, and in international network communications with multi-language configurations, where countries use different code pages.

To help ensure that character data is correctly represented and, if necessary, converted when you move data across computer platforms and international languages, DFSMS/MVS supports character data representation and conversion.

Data is represented by means of a Coded Character Set Identifier (CCSID). The CCSID is supplied by the user. For example, in a CCSID you identify the character data's encoding scheme, and which character set(s) and code page(s) are used. On DFSMS/MVS 1.5, data conversion is supported by BSAM and QSAM when using ISO/ANSI Version 4 tapes for data interchange with non-390 systems. Conversion can also be suppressed when required.

Managing Data Sets and Volumes

To manage the storage and placement of data sets, DFSMSdfp uses catalogs and volume tables of contents (VTOCs). DFSMSdfp also provides functions for managing data and storage, for gathering storage information, for sharing PDSEs, and for using VSAM RLS.

Cataloging Data Sets

A catalog describes data set attributes and records the location of a data set so that the data set can be retrieved without requiring the user to specify the data set's location. Multiple user catalogs contain information about user data sets, and a single master catalog contains entries for system data sets and user catalogs.

DFSMSdfp supports three types of catalogs, which can coexist on the same operating system: ICF catalogs, VSAM catalogs, and CVOLs. All data sets managed by the Storage Management Subsystem (SMS) must be cataloged in an ICF catalog.

VSAM catalogs and CVOLs will no longer provide access to cataloged data sets when the system date passes beyond the end of 1999; this support is provided strictly for compatibility. Consequently, you should plan to convert all VSAM catalogs and CVOLs to ICF catalogs.

Most installations depend on the availability of catalog facilities to run production job streams and to support online users. For maximum reliability and efficiency, all permanent data sets should be cataloged and all catalogs should be integrated catalog facility catalogs. See *DFSMS/MVS Managing Catalogs* for information on VSAM catalogs and CVOLS and how to convert to integrated catalogs.

Integrated Catalog Facility Catalogs

The integrated catalog facility catalog is a functional replacement for VSAM master and user catalogs and for CVOLs. Integrated catalog facility catalogs provide improvements over VSAM catalogs in reliability, recoverability, performance, usability, and DASD space management.

You can improve the performance of integrated catalog facility catalogs by placing the catalog in a data space cache. This improves the performance of the retrieval of integrated catalog facility catalogs by avoiding I/O to DASD and improves performance of shared integrated catalogs.

Unlike VSAM catalogs, integrated catalog facility catalogs do not own the volumes of their cataloged data sets. More than one catalog can have data sets residing on the same volume. See *DFSMS/MVS Managing Catalogs* for more information.

Considerations for a Multisystem Environment

If you share data or catalogs with lower-level systems, you must apply program temporary fixes (PTF) to prevent data contamination and catalog damage. See *DFSMS/MVS Program Directory* for details on which PTFs are required.

Cataloging Tape Libraries and Volumes

Tape libraries and the volumes they contain must be cataloged in a volume catalog. A volume catalog is an integrated catalog facility user catalog that can only contain entries for tape volumes and tape libraries. You can use access method services to define and maintain a volume catalog the same way you define and maintain an integrated catalog facility catalog, except that you cannot define aliases to a volume catalog.

There are two categories of volume catalogs: general and specific. Each system can have access to only one *general volume catalog* but can have access to many *specific volume catalogs*. A general volume catalog contains all the entries that define tape libraries as well as entries for tape volumes that are not cataloged in a specific volume catalog. Specific volume catalogs are volume catalogs that contain a specific group of volume entries based on the tape volume serial number.

A library entry contains information about a particular tape library, such as the name of the library and the number of empty slots in the library. There is exactly one library entry for each tape library. Library entries can only reside in the general volume catalog; they cannot reside in specific volume catalogs.

A volume entry contains information about a particular tape volume, such as the volume serial number, library name, and storage group name. There is exactly one volume entry for each tape volume.

Note: A tape volume can have two storage group names associated with it. System-managed OAM tape volumes containing objects will have both an OBJECT/OBJECT BACKUP storage group name and a TAPE storage group name associated with them. The catalog entry contains information concerning only the TAPE storage group name. Information concerning the OBJECT/OBJECT BACKUP storage group name associated with the tape volume is kept in a DB2 table.

Using Catalogs

You can use the access method services utility to create, maintain, back up, and recover integrated catalog facility catalogs. The multilevel alias facility improves performance of catalog selection by allowing you to define an alias that consists of multiple data set name qualifiers.

Access Method Services Tasks for Catalogs

You can use access method services to create and maintain integrated catalog facility catalogs. With access method services, you can:

- Create catalogs
- Convert VSAM catalogs and CVOLs to integrated catalog facility catalogs
- Define catalog entries
- Alter catalog characteristics
- Diagnose catalog errors
- Delete catalog entries
- Copy or merge catalog entries

- Back up catalogs
- Lock catalogs
- Recover catalogs
- List the entries in a catalog
- Examine the structural consistency of a catalog.

For information on the access method services commands and their parameters, see *DFSMS/MVS Access Method Services for ICF*.

Catalog Backup and Recovery

Access method services commands are available to copy an integrated catalog facility or VSAM catalog for backup and to recover the backup copy if necessary.

An integrated catalog facility catalog can be recovered with minimal impact on the surrounding subsystems. Subsystems like IMS and CICS that are oriented to an integrated catalog facility catalog do not have to be quiesced or terminated when the catalog must be recovered. DFSMSdfp automatically reorients the subsystems to the catalog after recovery.

Integrated catalog facility catalogs can be locked during the recovery period. This permits restricted access to the catalog so that recovery personnel can update the recovered catalog to the point at which a problem occurred or validate it while it remains inaccessible to other users. To lock a catalog, you must either have RACF or code an MVS System Authorization Facility (SAF) router exit routine. For further information on catalog backup and recovery, see *DFSMS/MVS Managing Catalogs*.

Catalog Selection

You can control catalog selection by defining aliases for user catalogs. When a user defines a new data set or accesses an existing data set, the system uses the high-level qualifiers of the data set name to select the required user catalog. With the multilevel alias facility, you can define aliases of up to four qualifiers. For more information, see *DFSMS/MVS Managing Catalogs* .

Sysplex Catalog Alias

All systems in a sysplex can use the same master catalog without JCL changes being required depending on which system a job runs.

In a sysplex environment with a shared master catalog, you can have an alias on one system, use a different catalog than the same alias uses on another system. For example, you can have an alias name of IMSCAT use the user catalog IMS.PRODCAT on System A and IMS.TESTCAT on System B. This is possible by the use of a system-specified variable for the user catalog name in the alias definition.

The specific user catalog name is resolved differently on each system. On each system the system-specified variable is set by the SYMDEF paramater in the LOADxx parmlib member.

For non-VSAM data sets symbolic substitution also allows a single alias name to refer to different data set names on different systems. For example, an alias name of SYS1.PL1LIB can resolve to SYS1.R4PL1LIB on System A and SYS1.R5PL1LIB on System B. This is possible by the use of a symbolic value for the data set in the alias definition.

The specific data set name is resolved differently on each system. For more information on using symbolic substitution for user catalogs and non-VSAM data sets, see *DFSMS/MVS Managing Catalogs*.

Enhanced Catalog Sharing

You can use the coupling facility to facilitate catalog sharing in a sysplex environment with the use of the enhanced catalog sharing method. You can use system symbols as part of an alias entry, which permits each member of the sysplex to resolve the alias to a different data set name or catalog.

Catalog Sharing in a Sysplex Environment: You can activate the enhanced catalog sharing method through the use of the MODIFY CATALOG operator command. For more information on the operator command and parameters, see *DFSMS/MVS Managing Catalogs*.

The integrity of the shared catalog is maintained by the use of a cache structure within the coupling facility. When all systems have access to the coupling facility, this eliminates a RESERVE, RELEASE, and I/O operation against the VVDS for each shared catalog request. All the sharing systems must be at DFSMS/MVS 1.5. See *DFSMS/MVS Managing Catalogs* for details on using enhanced catalog sharing.

Virtual Input/Output Data Sets

Temporary data sets can be handled with a function called virtual input/output (VIO). When your program needs temporary data sets defined with a VIO unit name, they are dynamically allocated in 4 KB physical blocks on the system's paging data sets. VIO stores an image of the tracks in virtual storage.

The advantage of virtual I/O data sets is that they eliminate some of the data management and I/O device overhead usually associated with temporary data sets. VIO uses DASD space more efficiently, and, because the I/O is handled by the paging mechanism of the system, it can be balanced with other I/O operations.

VIO data sets can be used with the BPAM, BSAM, QSAM, BDAM, and EXCP access methods. SMS can direct SMS-managed temporary data sets to VIO storage groups. Recovery processing is consistent with other kinds of temporary data sets, but VIO data sets are not eligible for deferred restart. PDSEs and VSAM data sets are not eligible for VIO.

Volume Table of Contents

A volume table of contents (VTOC) is a data set that resides on a DASD volume and describes the contents of that volume. The VTOC is composed of data set control blocks (DSCBs) that describe either the type and location of data sets on that volume or contiguous areas of unassigned space on the volume.

A VTOC can be indexed by a VTOC index data set residing on the same volume. The VTOC index provides direct access to the correct DSCB and manages free space information so that the number of I/O operations needed to obtain or release space on the volume is reduced. System-managed volumes require indexed VTOCs.

You can use the Device Support Facilities program, also known as ICKDSF, to create an indexed VTOC, convert an existing VTOC to indexed VTOC format, or expand a VTOC. You can use ISMF to generate a list of data set names from the VTOC.

ICKDSF is included in the base OS/390 offering. It is also a separately-orderable program (5655-257). For more information on ICKDSF, see *ICKDSF User's Guide and Reference*. For more information on the structure of the VTOC and its index, see *DFSMS/MVS DFSMSdfp Advanced Services*.

Note: DFSMS/MVS supports both indexed and nonindexed VTOCs, as does MVS/DFP Version 3.

System Data Administration

Certain components can be used to modify the data management capabilities of the operating system:

Direct Access Device Space Management (DADSM)
DADSM controls space allocation and deallocation on DASD volumes. DADSM is available to manage all DASD space except that under the control of a VSAM catalog. Although DADSM provides exit routines to help you control space allocation and enforce installation standards, you can improve storage management by using automatic class selection routines instead of the exits.

Execute Channel Program (EXCP)
You use the EXCP access technique to establish your own system for organizing, storing, and retrieving data. EXCP lets you tailor your data organization based on device characteristics, but such tailoring generally requires more work than using regular access methods and produces device type dependencies that might cause problems when migrating to a different device type. For further information, see *DFSMS/MVS DFSMSdfp Advanced Services*.

Disaster Recovery and Application Migration

Extended remote copy (XRC) is a combined 3990 and DFSMS/MVS disaster recovery and workload migration solution.

Using Remote Copy for Disaster Recovery

XRC is the optimal performance choice for shadowing your critical application volumes to a remote storage subsystem. XRC automatically sends copies of updated data to a remote recovery system with almost no impact to application system operations. To implement remote copy, an installation establishes two systems: an application system at one location and a remote, recovery system at another location. Each system has specific DASD that handles data that you have identified as remote copy-managed.

Once established, remote copy makes changes to your data on the remote DASD subsystem as you make those changes at your application location. If your application system fails, recovery involves a takeover by the recovery system, which may be located miles from your application system.

Using Remote Copy for Workload Migration

Although the primary purpose of extended remote copy is recovering data in the event of an unrecoverable error, XRC is also an efficient tool for moving, or migrating, data from one set of DASD volumes to another set with minimal impact to applications.

For more information, see *DFSMS/MVS Remote Copy Administrator's Guide and Reference*.

Backup, Recovery, and Space Management

ISMF helps you analyze and manage both data and storage interactively, using space management and backup and recovery functions provided by the DFSMSdss and DFSMShsm functional components of DFSMS/MVS.

DFSMSdss

DFSMSdss lets you copy, move, dump, and restore data sets and volumes for backup and recovery. DFSMSdss also relocates data set extents on a DASD volume to reduce or eliminate space fragmentation. It then prints a report about free space and other volume statistics to help you determine when to compress volumes or data sets and recover unused space. When your data resides on a DASD volume that supports the concurrent copy function, DFSMSdss can make a consistent backup or copy of your data while an application program uses your data concurrently. For more information on DFSMSdss, see Chapter 9, “Role of the Functional Component DFSMSdss” on page 83.

DFSMShsm

DFSMShsm lets you migrate data sets to different levels of storage according to how frequently they are needed. DFSMShsm also assists in backing up and recovering data sets and managing space on DASD volumes. If you are operating in an active SMS environment, DFSMShsm provides backup and space management services defined by the management class and storage group attributes and the automatic class selection routines. For more information on DFSMShsm, see Chapter 11, “Role of the Functional Component DFSMShsm” on page 95.

The ISMF aggregate group application allows the storage administrator to control DFSMShsm processing for the backup and recovery of groups of data sets that require concurrent action. See “Defining Aggregate Groups for Disaster Backup and Recovery” on page 32 for details.

DFSMSrmm

DFSMSrmm lets you move tape volumes between system-managed libraries and DFSMSrmm storage locations, and manage the retention of data sets and volumes.

DFSMSdfp Callable Services

DFSMSdfp allows your programs to use the services of the DFSMS Attribute Service. These services can be called by user programs written in assembler language and in the high-level languages supported by Systems Application Architecture (SAA).

The DFSMS attribute service supports these callable system services:

IGWABWO
You can use this service to read, set, and reset some fields for system-managed VSAM data sets in order to allow CICS VSAM File Control data sets to be backed up while they are open for update.

If you do not stop or quiesce an application to create backup copies of data sets that are open, the backups you create contain data that might be changing during the duration of the backup and, unlike concurrent copy, is not an image of the data at the time the backup is started. To prevent invalidation of backups of VSAM KSDSs due to control interval or control area splits, or addition of data at the end of the data set during the backup, use backup-while-open processing with concurrent copy.

IGWARLS
You can use this service to get information for a system-managed CICS VSAM sphere for which RLS attributes have been defined. IGWARLS tells you if the sphere was defined as recoverable, and whether recovery is pending. The recoverable/non-recoverable attribute of a data set determines whether CICS performs logging for it, and what level of sharing is allowed among applications seeking access to the data set.

IGWASMS
You can use this service to determine if a data set is system-managed. If the data set is system-managed, IGWASMS returns the names of any related SMS classes and indicates whether the data set is a PDSE.

IGWASYS
You can use this service to determine the version, release, and modification level of DFSMS/MVS and the status of SMS on your system.

IGWLSHR
You can use this service to determine the DFSMSdfp share attributes currently in use on the system. You can use this service to optimize PDSE access protocols. With support for concurrent sharing of a PDSE for output between multiple MVS systems, you can open a PDSE for OUTPUT for an extended period without locking out other INOUT or OUTPUT sharers of the PDSE.

Note: The three callable services IGWASYS, IGWASMS, and IGWABWO supersede IGWAQSMS, which is no longer available.

For more information on DFSMSdfp callable services, see *DFSMS/MVS DFSMSdfp Advanced Services*.

Collecting Storage and System Information

The access method services DCOLLECT command lets you collect information about data sets, volumes, and DFSMShsm-owned storage. With DCOLLECT, you can obtain information about space use, data set attributes, data sets residing on selected volumes and storage groups, volume statistics and information. You can also collect information for data sets migrated and backed-up by DFSMShsm and information and statistics for volumes and tapes managed by DFSMShsm to help you with both DASD and tape capacity planning. DCOLLECT provides the following information about your SMS configuration: construct attributes, SMS volume details, SMS base configuration, aggregate groups, optical drives and libraries, cache names, and ACS accounting information.

You can run DCOLLECT through ISMF. For more information on DCOLLECT, see the following publications:

- *DFSMS/MVS Access Method Services for ICF*
- *DFSMS/MVS DFSMSdfp Storage Administration Reference*,
- *MVS/ESA SML: Managing Data*

Sharing PDSEs Among MVS Systems

DFSMS/MVS allows users on different MVS systems to access the same PDSE simultaneously. A user who is sharing a PDSE can read existing members in the PDSE or create new members or new copies of existing members concurrently with other users on the same MVS system and on other MVS systems. Data integrity is still maintained because only one user at a time can modify a data set member without copying it.

Using VSAM Record-Level Sharing

VSAM record-level sharing (RLS) extends the DFSMS/MVS storage hierarchy to support a data sharing environment across multiple systems in a Parallel Sysplex. This support is designed primarily for VSAM data sets used by CICS Online Transaction Processing (OLTP) applications. VSAM RLS processing involves support from multiple products: CICS Transaction Server, CICSVR 2.3, and DFSMS/MVS 1.3 or higher.

VSAM RLS is a data set access mode that allows multiple address spaces, CICS application owning regions (AORs) on multiple MVS systems, and jobs to access data at the same time. With VSAM RLS, multiple CICS systems can directly access a shared VSAM data set, eliminating the need for function shipping between AORs and file owning regions (FORs).

CICS provides the logging, commit, and rollback functions for VSAM recoverable files; VSAM provides record-level serialization and cross-system caching; and CICSVR provides a data set recovery function. Whether a data set is recoverable or not determines the level of sharing allowed between applications. For example:

- Both CICS and non-CICS jobs can have concurrent read/write access to non-recoverable data sets.
- Non-CICS jobs can have read-only access to recoverable data sets, concurrent with read/write access by CICS jobs.

You use the LOG data set attribute to define a data set as recoverable or non-recoverable.

Note: Batch jobs that update recoverable data sets continue to use non-RLS mode (non-shared resource (NSR), local shared resource (LSR), global shared resource (GSR) access modes.) They cannot concurrently update any data set being accessed in RLS access mode.

VSAM RLS uses the coupling facility (CF) to perform data set level locking, record locking, and data caching. VSAM RLS uses the conditional write and cross-invalidate functions of the CF cache structure, thereby avoiding the need for control interval (CI) level locking.

VSAM RLS uses the CF caches as store-through caches. When a control interval of data is written, it is written to both the CF cache and to DASD. This ensures that problems occurring with a CF cache do not result in the loss of VSAM data.

To enable VSAM RLS, you must:

- Run all systems performing RLS as a Parallel Sysplex
- Define and activate at least two sharing control data sets (SHCDS), and one spare SHCDS for recovery purposes
- Define CF cache and lock structures to MVS, using the coupling facility resource manager (CFRM) policy, and to the SMS base configuration. See "Defining Use of the Coupling Facility for VSAM Record-Level Sharing" on page 35 for more information on the base configuration.
- Associate CF cache set names with storage class definitions, and write ACS routines to associate data sets with storage class definitions that map to CF cache structures. See "Managing Performance and Availability" on page 26 for more information on defining storage class definitions.
- Change the attributes for a data set to specify whether the data set is to be recoverable or non-recoverable.

See the following for more information on VSAM RLS:

- *DFSMS/MVS Planning for Installation*
- *DFSMS/MVS DFSMSdfp Storage Administration Reference*
- *DFSMS/MVS Using Data Sets*
- *CICS Transaction Server for OS/390 Migration Guide*
- *OS/390 Parallel Sysplex Hardware and Software Migration*
- *OS/390 Parallel Sysplex Application Migration*
- *CICS Recovery and Restart Guide*
- *CICSVR MVS/ESA V2R3 Implementation Guide*
- *CICSVR MVS/ESA V2R3 User's Guide and Reference*

Controlling Access to Data

Security is the ability to protect data processing resources from unauthorized access, alteration, or destruction. DFSMSdfp supports four methods of controlling access to sensitive data:

Resource Access Control Facility (RACF) protection
Authorized Program Facility (APF) protection
Access Method Services Cryptographic Option protection
Password protection facility.

DFSMS/MVS and B1 Security

A Trusted Computing Base, anchored by MVS/SP Version 3 Release 1.3 and including MVS/DFP Version 3 Release 1.1, was certified in September, 1990, by the U.S. Department of Defense National Computer Security Center (NCSC) as meeting B1 security criteria. The DFSMSdfp functional component of DFSMS/MVS, supported by MVS/ESA SP Version 4 and Version 5, has not been evaluated by the NCSC, but does contain the security functions previously contained in MVS/DFP Version 3 Release 1.1.

| However, a system does not meet the criteria for B1 security if the OS/390
| Network File System , the Distributed FileManager/MVS, or the Object Access Method (OAM) of the DFSMSdfp functional component are running, or if the functional components DFSMSdss, DFSMShsm, or DFSMSrmm are used. Objects managed by OAM are protected by DB2 security, rather than by RACF, and are not protected at the B1 security level. OAM also provides a user exit, CBRUXSAE, that be coded for use with OSREQ functions.

Resource Access Control Facility Protection

The IBM Resource Access Control Facility (RACF) controls access to data processing resources. All four functional components of DFSMS/MVS— DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm—support RACF. For more information on how the other functional components support RACF protection, see “Controlling Access to DFSMSdss Tasks and Data” on page 94, “RACF” on page 121, and Chapter 13, “Role of the Functional Component DFSMSrmm” on page 123.

You should use RACF protection instead of data set password protection because RACF protection is easier to use and offers greater system resource security. In addition, SMS ignores data set passwords. You can protect non-SMS-managed data sets through RACF, password protection, or both, although RACF is the preferred method. If RACF protection is applied to a data set that is password protected, password protection is bypassed and access is controlled solely through RACF.

RACF retains information about users, resources, and access authorities in profiles stored in a special RACF database. *Discrete profiles* contain security information about a single data set or other resource. *Generic profiles* contain security information about one or more data sets or other resources that might have similar characteristics and therefore require a similar level of protection. Thus, generic profiles make it easy for you to protect multiple data sets with similar security requirements, without having to specify the security requirements individually for each data set.

To verify access, various parts of the operating system, such as SMS, DADSM, Open/Close/EOV, Checkpoint/Restart, IEHMOVE, and the integrated catalog facility, use RACF to automatically check all data sets on DASD for generic and discrete profiles. This is referred to as *RACF always call* because RACF is always called to verify access to a data set. For VSAM data sets, DFSMS/MVS checks only the cluster's profile for access authorization to any of the cluster's components. Although you can have discrete profiles defined for the data and index components of a data set, DFSMS/MVS does not check them. If the cluster does not have a separate profile, you must consider the individual components unprotected.

With RACF, erase-on-scratch can be controlled by RACF options and data set profiles for both non-VSAM data sets and integrated catalog facility cataloged VSAM data sets. Erase-on-scratch prevents unauthorized access to sensitive DASD data by automatically erasing a data set when it is scratched. Open/Close/EOV support for RACF includes protecting tape data sets, non-labeled (NL) tapes, and tapes using bypass label processing (BLP). This support provides protection for both DASD and tape data sets under the same generic profile.

The storage administrator can use the RACF program control facility to establish authorization levels for all ISMF functions or individual applications, functions, line operators, and commands. Users who are denied access because of insufficient authority are notified by an informational message.

An installation can also use RACF to control the use of storage and management classes. The default for use of storage and management class is the RACF RESOWNER value, which is based on the high-level qualifier of the data set name. Instead, you can use management and storage class based on the allocator of the data set. This prevents the problems that can occur with restoring or recalling data sets which have a protected storage class and management class, and which are owned by users whose user or group IDs have been revoked.

See *DFSMS/MVS DFSMSdfp Storage Administration Reference* for details. For more information on RACF protection, see:

OS/390 Security Server (RACF) Introduction
OS/390 Security Server (RACF) General User's Guide
MVS/ESA SML: Managing Data

Authorized Program Facility (APF) Authorization

The use of sensitive system services and resources can be restricted to APF-authorized system and user programs. An APF-authorized program can bypass all security and protection. Some DFSMS/MVS functions (such as the access method services CNVTCAT command) are stored in authorized libraries and can be used only by users that have the proper level of RACF authorization. For more information on APF authorization, see *OS/390 MVS Authorized Assembler Services Guide*.

Cryptographic Protection

The cryptographic option of access method services can help protect sensitive data that is copied. Data can be encrypted using the access method services REPRO command with the ENCIPHER option. The data remains encrypted until the REPRO command with the DECIPHER option is used to decipher the data with the correct key.

You can use cryptographic protection if you have the IBM Programmed Cryptographic Facility (5740-XY5) licensed program (with its prerequisite hardware) installed on your system. The Programmed Cryptographic Facility method of enciphering data conforms to the Data Encryption Standard (DES) of the United States National Institute of Standards and Technology (NIST). For more information on protecting off-line data with the cryptographic option, see *DFSMS/MVS Using Data Sets*.

The Integrated Cryptographic Service Facility/MVS (5685-051) is also available for water-cooled ES/9000 processors and specific other models. This licensed program works with the Integrated Cryptographic Feature to provide high-performance processing of large amounts of encrypted data. For more information, see *Integrated Cryptographic Service Facility/MVS General Information*.

Data Set Password Protection

You should use RACF to protect your data. As described in "Resource Access Control Facility Protection" on page 57, RACF protection is much more robust than data set password protection and also easier to use.

Passwords are ignored for SMS-managed data sets and all other ICF-cataloged data sets. Passwords will not be used to determine a user's authorization to define, update, delete or otherwise access or change entries in ICF catalogs.

For more information, see *DFSMS/MVS DFSMSdfp Advanced Services*.

Chapter 6. Program Management with DFSMSdfp

DFSMS/MVS provides functions to create, load, modify, list, read, transport, and copy executable programs. DFSMS/MVS offers two primary linking and loading programs: the *program management binder* and the *program management loader*.

The binder extends the functions formerly provided by the MVS/DFP linkage editor and batch loader. The binder's enhancements include support for an executable unit called a *program object*, which has many of the features of a load module. To support the new program object, the loader adds new loading mechanisms to the program fetch function.

The binder also includes functions previously handled by the C Pre-Link Utility, so that an intermediate C Pre-Link Utility step between the compilation and binding of C and C++ modules is no longer required. Additionally, CSECTs within a C module can now be partially updated; previously, this was only possible from object module input, which could not be a load module or a program object.

The binder and loader contain many functional and usability enhancements compared to the MVS/DFP linkage editor, batch loader, and program fetch components. Note that IBM plans no further enhancements to the linkage editor, batch loader or program fetch.

For more information on program management with DFSMS/MVS, see *DFSMS/MVS Program Management*.

This chapter describes how the program management binder and program management loader are used to manage load modules and program objects.

Creating and Executing Load Modules and Program Objects

The binder and loader are used primarily to convert object modules into executable programs, store them in program libraries, and load them into virtual storage for execution.

Figure 12 on page 62 shows how programs can be prepared for execution using the binder and the loader.

© Copyright IBM Corp. 1980, 1999

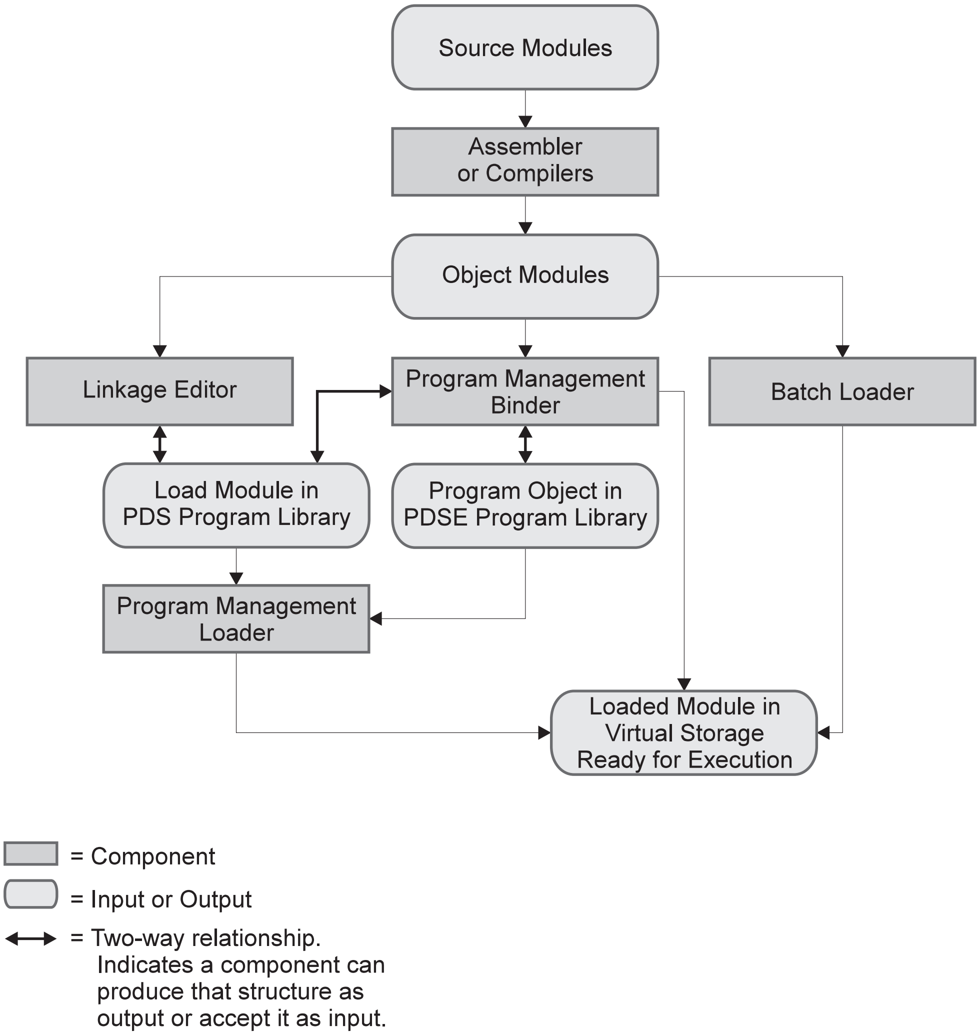

Source Modules
Assembler or Compilers
Object Modules
Linkage Editor
Program Management Binder
Batch Loader
Load Module in PDS Program Library
Program Object in PDSE Program Library
Program Management Loader
Loaded Module in Virtual Storage Ready for Execution
= Component
= Input or Output
= Two-way relationship. Indicates a component can produce that structure as output or accept it as input.

Figure 12. Creating and Loading Programs

Converting Modules and Program Objects

Source modules are units of code written in a specific programming language. They are processed by a compiler or by a language translator into *object modules*, units of machine code that can be relocated but not executed. To run an object module, it must be processed by a linker, such as the binder or the linkage editor.

The binder converts the output of the translator or compiler into either a *program object* or a *load module*; the linkage editor converts the same output into a load module. Both are executable forms of a program that can be moved directly into storage by the binder and processed immediately, or stored in a program library and loaded from this library by either the loader or the batch loader.

In addition to traditional object modules, the binder accepts an extended object module format that supports long external names up to 1024 bytes, multiple text classes, and embedded ADATA. The binder also accepts reentrant C and C++ modules, eliminating the need for the C Pre-linker.

Load modules are stored in partitioned data set (PDS) program libraries; program objects must be stored in a partitioned data set extended (PDSE) program library, or in a OS/390 UNIX file used by OS/390 UNIX. Program objects eliminate many of the restrictions of load modules. For example, a program object can have a text size of up to 1 GB, whereas the text size of a load module cannot exceed 16 MB. Additionally, the size of text records within program objects is fixed, eliminating the need to reblock when you copy programs between devices with different track sizes.

You can use the binder to process object modules convert load modules or program objects as follows:

- Convert object or load modules into a program object
- Convert extended object modules into a program object
- Convert object modules, load modules, or program objects into a load module and store it in a PDS program library. This is equivalent to what the linkage editor can do with object and load modules.
- Convert object or load modules into an executable program in virtual storage and pass control directly to the completed program. This is equivalent to what the batch loader can do with object and load modules.

You can also use the IEBCOPY utility to convert load modules in a PDS into program objects in a PDSE, or program objects in a PDSE into load modules in a PDS. See “Using Utilities for Program Management” on page 65 for more information on IEBCOPY.

Linking Modules and Program Objects

The binder performs all of the functions of the linkage editor. It *link-edits*, or combines and edits individual object modules or load modules to produce a single load module that is stored in a PDS program library until it is needed. Additionally, the binder can link-edit or *bind* modules into a program object that can be stored in a PDSE program library. When a member of a program library is needed, the loader brings it into virtual storage and prepares it for execution.

As input, the binder accepts a combination of programs divided into multiple object modules, load modules, and program objects. Control statements specify how to combine these modules into one or more load modules or program objects with contiguous virtual storage addresses. The binder creates programs in both the 24-bit and 31-bit addressing range. Each program module can be separately assembled or compiled, so that changes made to a program require only the changed modules to be recompiled or reassembled. The binder can also create overlay load modules or program objects.

The binder supports a generalized dynamic linking/loading capability which can be used by any language. This capability results in the deferred binding of some functions and variables until the application is run. Dynamic link libraries (DLLs) and DLL modules are implemented as PDSE libraries and program objects.

The binder relaxes or eliminates many restrictions of the linkage editor. The binder removes the linkage editor's limit of 64 aliases, allowing a load module or program object to have as many aliases as desired. The binder accepts any system-supported block size for the primary (SYSLIN) input data set, eliminating the linkage editor's maximum block size limit of 3200 bytes. The binder also does not restrict the number of external names, whereas the linkage editor set a limit of 32767 names.

The binder supports a length of 1024-byte for external names and aliases when binding external labels and references for program objects. Primary member names are still limited to eight bytes.

Loading Modules and Program Objects in Virtual Storage

You can use either the binder or the loader to load modules into virtual storage.

Using the Binder

The binder performs all of the functions of the batch loader, combining basic linking and loading functions in one job step.

Like the batch loader, the binder can read object modules and load modules from PDS program libraries into virtual storage, relocate the address constants, and pass control directly to the program upon completion. The binder also accepts program objects from PDSE program libraries as input. Just as the batch loader cannot store its output in a program library, neither does the binder when it is invoked to prepare input for execution.

You can use the binder like the batch loader for efficient, one-time loading of modules that do not require special processing services. Because the binder can perform the functions of the batch loader and process a job in a single job step, modules can be processed by the binder in virtual storage, with reduced linking and loading time.

Using the Loader

The program management loader incorporates all of the functions of the MVS/DFP program fetch component and adds support for loading program objects, including distributed loading of multi-part program objects.

The loader reads load modules into virtual storage and prepares them for execution. It adjusts any address constants in the load module to point to the appropriate locations in virtual storage, which can be in either the 24- or 31-bit addressing range.

You can use the loader to load program objects. The loader supports five modes for loading a program object into virtual storage from a PDSE program library. The loader selects a mode based on module characteristics and parameters you specify to the binder. For example, in move mode the loader moves the text into virtual storage and relocates the address constants prior to the program receiving control. In page mode, the loader “page loads” text pages into storage as that text is referenced and relocates address constants of the text pages as the text pages are loaded. Using page mode loading enables you to begin executing large programs without waiting for the entire program to be read into storage and relocated.

If you created a multi-part program object during binding, you can use the loader to load those parts below and above the 16 MB line, and to pass control to the designated entry point.

The loader supports the dynamic loading of DLLs. This allows external references in one module to other modules to be resolved by loading the appropriate module at program load time, or when the module is referenced. This capability is also available on other platforms, which should facilitate application migration among platforms.

Loading Modules in a Shared Environment

Support for loading modules in a shared environment is provided by the Library LookAside (LLA) method and the Link Pack Area (LPA).

The LPA is a system-defined area reserved in all address spaces for program sharing. It contains frequently used, reenterable programs from system-designated libraries and is loaded during system startup. Loading a system, subsystem, or application program in the LPA can greatly reduce its load time. OS/390 includes an extended LPA function, Dynamic LPA, which allows PDSEs, and therefore program objects and DLLs, to be included in LPA. DFSMS/MVS 1.4 supports this functions. Before this, only PDSs, and therefore load modules, could be part of LPA.

LLA supports both the caching of PDS and PDSE program directories, as well as the caching of load modules and program objects (when they are referenced frequently enough). Note that LLA processing can be explicitly bypassed using DESERV to get the directory entry. An option on the DESERV macro lets the caller tell the system to get the copy of the module from DASD and not from LLA. The binder takes advantage of this to bypass LLA processing.

Using Utilities for Program Management

Although utility programs primarily support data management tasks, you can also use the IEBCOPY, IEHPROGM, IEHLIST and IEWTPORT utilities to support program management tasks.

For more information on IEBCOPY, IEHPROGM, and IEHLIST, see *DFSMS/MVS Utilities*.

For more information on IEWTPORT, see *DFSMS/MVS Program Management*.

IEBCOPY Utility

You can use the IEBCOPY utility program to alter relocation dictionary (RLD) counts of load modules in place, to copy and reblock load modules, to copy program objects, to convert load modules into program objects, or to convert program objects into load modules.

The COPYMOD option reblocks load modules to a block size best suited for the device to which you are copying the data set, allowing you to maximize DASD track utilization and reduce the time it takes to load a program into virtual storage. You do not need to alter RLD counts for program objects, nor do you need to use the COPYMOD option to change the block size of a program object library.

When you use IEBCOPY to copy a load module from a PDS library into a PDSE library, IEBCOPY automatically converts the new copy into the appropriate format for the target program library. Other utilities and application programs can use the binder's interactive call interface to include a module from one library and save the module in a second library.

Likewise, you can use IEBCOPY to convert program objects into load modules when you copy them into a PDS program library. However, you cannot convert a program object into a load module and store it in a PDS library if the program object exceeds the limitations of load modules. For example, you cannot convert a program object into a load module if the text length is greater than 16 MB or if it contains external names longer than eight bytes.

IEHPROGM Utility

You can use the IEHPROGM utility or TSO commands to delete or rename load modules, program objects, or their aliases. If you delete the primary name of a load module, that name is deleted and the aliases are left. If you delete the primary name of a program object, that name and all its aliases are deleted. You cannot use IEHPROGM or TSO commands to rename or delete alias names longer than eight bytes.

IEHLIST Utility

You can use the IEHLIST utility or TSO commands to list entries in the directory of one or more PDS or PDSE program libraries. IEHLIST can list up to 10 PDS or PDSE directories at a time in an edited or unedited format.

IEWTPORT Utility

You can use the IEWTPORT utility to prepare a program object for access on a system where the binder is not installed. IEWTPORT converts the program object into a nonexecutable, open format, called a *transportable program*. You can transfer this program to other systems where the binder is not available.

IEWTPORT also converts transportable programs into program object format, so that program objects can be loaded, bound, and processed.

Using Service Aids for Program Management

Service aids are programs designed to help system programmers and IBM program support representatives diagnose and repair errors in system or application programs. You can use the AMBLIST and AMASPZAP service aids to perform some program management tasks. Both AMBLIST and AMASZAP support program objects, long names up to 1024 bytes, and multiple text classes.

See *OS/390 MVS Diagnosis: Tools and Service Aids* for more details on using AMBLIST and AMASPZAP.

Listing Module Information with AMBLIST

You can use the AMBLIST service aid program to print formatted listings of modules and system storage areas to aid in problem diagnosis. AMBLIST does not provide support for program objects in OS/390 UNIX files.

You can use AMBLIST to print listings of load modules and program objects showing:

- Module attributes
- The contents of the various classes of data contained in a module, including SYM records, IDR records, external symbols (ESD entries), text and relocation dictionary entries (RLD entries)
- A module map or cross reference for a module
- The aliases of a module, including the attributes of the aliases

You can print listings of the modified link pack area (MLPA), fixed link pack area (FLPA), pageable link pack area (PLPA), and their extended areas in virtual storage together or separately.

Inspecting and Modifying Load Modules with AMASPZAP

The AMASPZAP service aid, also called SPZAP or Superzap, helps you dynamically update or dump programs and data sets. You can use AMASPZAP to inspect and modify instructions in any load module or program object in a program library or OS/390 UNIX file. AMASPZAP can be used to dump a load module or program object in a program library or OS/390 UNIX file. You can also use AMASPZAP to update the system status index in the directory entry for any load module or program object.

Chapter 7. Device Management with DFSMSdfp

DFSMS/MVS device management components define input and output devices to the operating system and help control the operation of these devices. DFSMS/MVS provides programming support to help you best use the standard and optional features of your devices. You can use some DFSMS/MVS functions with many different device types, but most apply specifically to one type or one family of devices.

The sections that follow discuss general device support as well as the tasks that DFSMS/MVS can perform to support each device type.

DFSMS/MVS Device Support

This section introduces the operating modes in which devices can operate, provides a list of the most common IBM I/O devices supported by DFSMS/MVS, MVS/ESA SP, and OS/390, discusses how to define new devices to DFSMS/MVS, and discusses channel and device support.

Operating Modes

Most devices attached to MVS operate in *full—function* mode; that is, all features on the device are compatible with and usable on the operating system. Some of these features include:

- DASD devices:
  - Dynamic path reconnection
  - Extended count-key-data operation
  - Caching and cache related facilities
- Tape devices:
  - Cartridge stack loading
  - Data compaction

Some devices also operate in *compatibility* mode, which lets you simulate the function of another device or model. Compatibility mode causes the device to function like a different device of the same type, ignoring some or all of the additional features that the device might have. This allows you to migrate between devices with minimal impact on programs that have device dependencies.

Common Supported IBM I/O Devices

Figure 13 lists the most common IBM I/O devices supported by DFSMS/MVS, MVS/ESA SP, and OS/390.

Note: Some devices might require a specific level of hardware maintenance to operate correctly on an MVS system. In addition, DFSMS/MVS software support of some new hardware devices might require the installation of program temporary fixes (PTFs).

See the REmote Technical Assistance and Information Network Program Services Period (RETAIN PSP) Hardware Install Indexes to identify any required PTFs.

© Copyright IBM Corp. 1980, 1999

Figure 13 (Page 1 of 2). IBM I/O Devices and Subsystems Supported by OS/390

Direct Access Storage

- 3380 Direct Access Storage[1]
- 3390 Direct Access Storage[1]
  - Models 1, 2, 3, and 9
- 9340 Direct Access Storage[2]
  - 9341—9345 Direct Access Storage
  - 9343—9345 Direct Access Storage
- 9396 Ramac Scalable Array Storage
- 9397 Ramac Electronic Array Storage
- RAMAC Array Device[11]
- 9391 RAMAC Array Subsystem[11]

Storage Control Unit

- 3880 Storage Control
  - Model 3[14]
- 3990 Storage Control
  - Models 1 and 2[1]

Cache Storage Control Unit

- 3880 Storage Control
  - Model 23 with 3880 AJ4/AK4 Attachment (Feature 3010)
- 3990 Storage Control
  - Models 3 and 6[1] [11] [12]
- 9393 RAMAC Virtual Array Storage Subsystem
  - Model xx2[15]

Console

- 3180 Display Station
  - Models 140 and 145
- 3205 Color Display Station
  - Model 100
- 3206 Display Station
  - Models 100 and 110
- 3251 Display Station
- 3277 Display Station
  - Models 1 and 2
- 3278 Display Station
  - Models 1, 2, 2A, 3, and 4
- 3279 Color Display Station
  - Models 2A, 2B, 2C, 3A, and 3B
- 5080 High Function Graphics System

Optical

- 9246 Optical Library Unit RPQ# 8B6001[10]
- 9247 Optical Disk Drive RPQ# 8B6003[10]
- Expand 9246 to Four Drives RPQ# 8B6002[10]
- MVS/ESA Direct Optical Attachment RPQ# 8B6004[10]
- 3995 Optical Library Dataserver
  - Models 11x, 13x, 15x, C1x, C3x
- 3995 Optical Library Drive
  - Models–SWx
- S/370 and S/390 Optical Media Attach/2[13]

Tape

- 2440 Magnetic Tape Subsystem
  - Models A01, A02, B01, and J02[3]
- 3420 Magnetic Tape Unit
  - Models 3, 4, 5, 6, 7, and 8
- 3422 Magnetic Tape Subsystem
- 3423 Device[13]
- 3424 Magnetic Tape Subsystem[4]
- 3430 Magnetic Tape Subsystem
- 3480 Magnetic Tape Subsystem
- 3490 Magnetic Tape Subsystem[5]
  - Base models
    - Models A01, A02, B02, and B04
  - Enhanced capability models
    - Models A10, A20, B20, B40, C10, C11, C1A, C22, C2A, D41, and D42
- 3494 Tape Library Dataserver
  - Model L10
- 3494 Virtual Tape Server
  - Models B16, B18
- 3495 Tape Library Dataserver
  - Automated
    - Models L20, L30, L40, and L50
  - Manual
    - M10
- 3590 Magnetic Tape Subsystem
- 3590-1 Magnetic Tape Subsystem

Printer

- 1403 Printer
  - Models 2, 7, and N1
- 3203 Printer
  - Model 5
- 3211 Printer
- 3262 Line Printer
  - Model 5[6]
- 3284 Printer
  - Models 1 and 2
- 3286 Printer
  - Models 1 and 2
- 3800 Printing Subsystem
  - Models 3, 6, 8[7] and 1
- 3812 Page Printer
  - Model 2[7]
- 3816 Page Printer
  - Models 01S, and 01D[7]
- 3820 Page Printer[8]
- 3825 Page Printer[8] [9]
- 3827 Page Printer[8] [9]
- 3828 Advanced Function MICR Printer[8] [9]
- 3835 Page Printer[8] [9]
- 3900 Advanced Function Printer[8] [9]
- 4028 LaserPrinter
  - Model NS1[7]
- 4245 Line Printer
  - Models 01, 12, and 20
- 4248 Impact Line Printer
  - Models 1 and 2
- 6262 Impact Line Printer
  - Models 014, and 022[6]

Other

- 2501 Card Reader
  - Models B1 and B2
- 2540 Card Read Punch
  - Models 1 and 2
- 3505 Card Reader
- 3525 Card Punch
- 37xx Communications Controllers
- 3838 Array Processor
- 3848 Cryptographic Unit
- 3890 Document Processor

Figure 13 (Page 2 of 2). IBM I/O Devices and Subsystems Supported by OS/390

Note:

1. Most models of the IBM 3380 Direct Access Storage and all models of the IBM 3390 Direct Access Storage can operate with Enterprise Systems Connection (ESCON) channels when attached to a 3990 Storage Control Model 2, 3, or 6 with ESCON capability.
2. All models of the IBM 9345 Direct Access Storage Devices mounted in an IBM 9343—9345 Model D04 cabinet can operate with ESCON channels.
3. These models of the IBM 2440 Magnetic Tape Subsystem are supported in 3420-4 compatibility mode.
4. The 3424 Magnetic Tape Unit is available only in Brazil, S.A.
5. All models of the IBM 3490 Magnetic Tape Subsystem can operate with ESCON channels if an ESCON adapter is installed.
6. This printer must be initialized to the system as 4248.
7. This printer requires the use of PSF for OS/390 if you want it to run as an Advanced Function Printer (AFP*). PSF for OS/390 is a separate product.
8. This printer requires the use of PSF for OS/390, which is a separate product.
9. This printer must be defined to the system as AFP1.
10. This feature or device is supported by the object access method.
11. All RAMAC subsystems, and the 3990 Storage Control Models 3 and 6 support SAM striping and SAM and VSAM KSDS compression.
12. The 3990 Storage Control Models 3 and 6 Extended Platform support concurrent copy.
13. The IBM S/370 and the S/390 Optical Media Attach/2 products emulate an IBM 3422 Magnetic Tape Subsystem, supporting both 1600 and 6250 bpi formats. They are defined to the system as 3423 tape devices.
14. The 3880 Storage Control Model 3 can support the 3380 AJ4/AK4 Attachment (Feature 3005).
15. The RAMAC Virtual Array supports both 3380 and 3390 device types in single, double, and triple capacity volume sizes.

Defining New Devices

You can define new devices to the system by using interactive panels with the Hardware Configuration Definition (HCD) program. HCD uses the dynamic I/O capabilities of MVS to change configuration definitions dynamically without requiring an initial program load (IPL) or hard power-on reset.

Although devices that are in tape libraries automated by an IBM 3494 or IBM 3495 Automated Tape Library Dataserver are automatically defined to the operating system by the dataserver, you should use HCD to ensure that the devices are set properly if they are off-line during the system IPL.

For information on the Hardware Configuration Definition, see *OS/390 HCD Planning* .

Channel and Device Support

The specific number of devices you can attach to your system depends on the hardware configuration of your processor and I/O devices, and the virtual storage below 16 MB. MVS/ESA and OS/390 architecture let you have as many as 65 536 device numbers, each with eight access paths. The device number is a four hexadecimal digit number, so that 4096 is no longer the limit for device addresses, as it was for previous releases.

With MVS/ESA SP 5.2 and OS/390, when you define DASD, tape, and optical devices to your system, DFSMS/MVS can place the UCBs representing those devices in 31-bit storage above 16 MB. This reduces the amount of below 16 MB common storage used by the system and provides you with an opportunity to define more devices to your system, needed to support large processors and Parallel Sysplexes. DFSMS/MVS maintains compatibility with 24-bit UCB addresses through the use of captured UCBs. DFSMS/MVS also supports access to UCBs above the 16 MB line to let subsystems like CICS and IMS take advantage of allocating data sets without capturing the UCBs.

DASD

DFSMS/MVS provides programming support for DASD through the Interactive Storage Management Facility (ISMF) and through Device Console Services. Additionally, you can use the Device Support Facilities program, also known as ICKDSF, to supplement programming support.

Interactive Storage Management Facility

ISMF helps you analyze and manage both data and DASD storage interactively. ISMF is an Interactive System Productivity Facility (ISPF) application that uses the space management and backup and recovery functions provided by DFSMShsm and DFSMSdss.

ISMF provides interactive interfaces for the access method services cache commands. It also provides the interactive interface for managing Storage Management Subsystem (SMS) configurations. You can use ISMF to define, validate, translate, and activate source control data sets (SCDSs). You can define SMS classes and groups and write automatic class selection routines. ISMF provides an interactive interface for submitting jobs to ICKDSF.

Device Console Services

Device Console Services commands display the logical status of DASD devices. The SET SMS, SETSMS, and ISMF ACTIVATE commands allow you to activate SMS or change its execution options. Three other commands, DEVSERV, DISPLAY, and VARY, support the tasks of checking, displaying and changing the status of the SMS configuration. For more information on DASD console commands, see *OS/390 MVS System Commands*.

Device Support Facilities (ICKDSF)

The Device Support Facilities program (ICKDSF) initializes DASD volumes, converts SMS-managed volumes, recovers data from defective tracks, and assigns alternate tracks. You can also use it to change a volume from a non-indexed VTOC to an indexed VTOC, or from an indexed VTOC to a non-indexed VTOC. Although not part of DFSMS/MVS, the Device Support Facilities program is an important related tool for DASD management. For more information, see *ICKDSF User's Guide and Reference*.

Storage Control Units

DFSMS/MVS offers programming support for cache devices. For detailed information on cache devices, see *3990 Planning, Installation, and Storage Administration Guide*.

LISTDATA, BINDDATA, and SETCACHE commands

These access method services commands control the caching and paging subsystems for the IBM 3880 Storage Control Models 13 and 23, and for the IBM 3990 Storage Control Model 3 or 6. Some commands are not supported for all models.

Using Concurrent Copy

System-managed data sets use storage class to specify availability and performance requirements. When a data set is created, the system uses these requirements to determine the device to which the data set should be allocated.

You can use the *accessibility* attribute of the storage class to specify if you want the data set allocated on a DASD that supports concurrent copy. Concurrent copy lets you make a consistent backup or copy of data concurrent with normal application program processing. In some cases you might need to quiesce the application before you start a backup or copy operation using concurrent copy.

When you allocate a data set, you can use the accessibility attribute of your storage class to specify whether the system must allocate the data onto a DASD supporting concurrent copy or end the allocation unsuccessfully, whether it should allocate the data onto another device if a concurrent copy device is not available, or whether it should allocate the data onto a DASD that does not support concurrent copy.

Using Remote Copy

The remote copy function relies on IBM 3990, RAMAC Storage Subsystems, or 9393 RAMAC Virtual Array Storage Subsystems and DFSMSdfp. Remote copy supports all DASD data needed for application recovery through interaction with the storage controls at your main application location and at your remote recovery location. Each location has specific DASD that handles data that you have identified as remote copy-managed.

In case of an unrecoverable error at the application location, data is recovered from the recovery storage system.

Remote copy also offers a way to easily migrate your peer-to-peer remote copy (PPRC) data from one DASD volume to another with the P/DAS function. See “Move Your Peer-to-Peer Remote Copy (PPRC) Data Easily with P/DAS” on page 20 for more information.

Using Continuous Availability

When you allocate a data set and specify the *continuous availability* attribute in your storage class, SMS allocates the data set to a dual copy device (supported by the 3990 Model 3 or 6), or to a RAID protected device (support by all RAMAC subsystems). If continuous availability is required and neither device is available, the allocation is unsuccessful.

Using Extended Format Data Sets

When you allocate a data set and specify a data set name type of EXTENDED, SMS allocates the data set to a system-managed DASD volume on a 3990 Model 3, Model 6, or any RAMAC subsystem. You can specify whether the data must be allocated on a DASD supporting extended format data sets, or whether the data should be allocated on another device if a device supporting extended format data sets is not available.

Caching on the IBM 3990 Model 3 or 6

SMS can select the 3990 Model 3 or 6 to provide caching for both read and write operations. The DASD fast write feature of the 3990 Model 3 or 6 provides caching for write operations.

Caching of system-managed data sets on a 3990 Model 3 or 6 is managed at the data set level. You can use access method services or the ISMF volume application to enable caching and DASD fast write at the volume level. You can define storage classes to indicate at the data set level which data sets:

- Must use cache
- Must never use cache
- Can use cache at the discretion of SMS

This interaction of the IBM 3990 Model 3 or 6 working together with DFSMS/MVS lets you:

- Control the amount of data to be cached
- Provide fast I/O response time to selected data sets
- Make full use of DASD capacity by mixing cached and noncached data sets on the same volume
- Direct SMS to adjust the amount of data being cached, based on current cache use

Caching on the IBM 3880 Model 23
SMS can select the 3880 Model 23 to provide caching for read operations. Caching for a system-managed volume on a 3880 Model 23 is enabled at the volume level, using access method services or the ISMF Volume Application.

Caching on RAMAC Subsystems
Caching is always active on the 9393 RAMAC Virtual Array Storage (RVA) and 9396 RAMAC Scalable Array Storage (RSA). The 9397 RAMAC Electronic Array Storage (REA) is a cache-only subsystem.

Magnetic Tape Volumes and Libraries

DFSMS/MVS provides the following programming support for magnetic tape devices:

Tape label support
DFSMS/MVS can process magnetic tapes with IBM standard, nonstandard, and ISO/ANSI labels. ISO/ANSI support processes magnetic tape labels and files that comply with international, American, and federal tape standards[1] as interpreted by IBM. For more information on tape label support, see *DFSMS/MVS Using Magnetic Tapes* .

IEHINITT utility
IEHINITT writes IBM standard or ISO/ANSI labels on tape volumes. [1] You can use DFSMSrmm's EDGINERS utility to initialize your tape volumes instead of using IEHINITT. EDGINERS provides several functions that IEHINITT does not. For more information on IEHINITT, see *DFSMS/MVS Utilities*. For more information on EDGINERS, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

IFHSTATR utility
IFHSTATR formats and prints information about tape volume errors from type 21 system management facilities (SMF) records. For more information on IFHSTATR, see *DFSMS/MVS Utilities*.

MSGDISP, SYNCDEV, NOTE, and POINT macros
These macros control operation of the IBM magnetic tape subsystems that use tape cartridges. The MSGDISP macro displays program messages on the

[1] International Organization for Standardization (ISO) 1001-1979, level 4; American National Standards Institute (ANSI) X3.27-1978, level 4. Tapes meeting these standards are called ISO/ANSI Version 3 tapes.

display on the tape unit. The SYNCDEV macro controls data synchronization. The NOTE and POINT macros control high-speed searching. For more information on these macros, see *DFSMS/MVS Macro Instructions for Data Sets* and *DFSMS/MVS DFSMSdfp Advanced Services*.

Device Console Services

Device Console Services commands display the logical status of tape devices. For example, the DEVSERV command tests all paths to a device and displays information about the logical status of each path on the system console. The operator can use this information to diagnose problems in the I/O subsystem.

Many device console commands simplify the task for operators with automated tape libraries. Use the DISPLAY command to request information about the status of a device. Use the MOUNT command to mount a volume and verify that the requested volume has been mounted. Use the EJECT command to eject a tape. Use the VARY command to vary a device online or offline. Use the LIBRARY EJECT command to eject a cartridge from a library and remove the catalog entry for the cartridge. Use the ACTIVATE command to inform the system of the existence of a library string.

For more information on console commands, see *OS/390 MVS System Commands*.

Tape block count checking

Block count checking compares the number of tape blocks calculated by the tape subsystem with the number of blocks calculated by the access method and detects discrepancies between the two. Block counts are checked when tapes are both read and written to improve tape reliability. Block counts are checked for all magnetic tape devices that use block identifiers (such as the 3490). Block counts are checked for labeled and unlabeled tapes.

Interactive Storage Management Facility

ISMF lets you define tape libraries and storage groups interactively. You specify the attributes related to the library or storage group, and the information is stored in an appropriate volume catalog. You can also use ISMF to create, save, and restore lists of tape volumes in libraries.

You can also use ISMF to define SMS classes, groups, and automatic class selection routines.

Library Control Services

The library control system provides several macros that allow other components to access and manipulate the information stored in tape volume catalogs. The services provided by these macros include verifying that the volume serial specified can be mounted on the device specified, and changing the status of an input volume from private to scratch, or from scratch to private.

Printers

Although most printers require the PSF for OS/390 product for full-function support, DFSMS/MVS continues to support the IEBIMAGE utility and the IGGUCSIT and SETPRT macros for older printers.

- For more information on IEBIMAGE, see *DFSMS/MVS Utilities*.
- For more information on IGGUCSIT, see *DFSMS/MVS DFSMSdfp Advanced Services*.

- For more information on SETPRT, see *DFSMS/MVS Macro Instructions for Data Sets*.
- For more information on PSF for OS/390, see *Advanced Function Printing: Printer Summary*.

Object Support

DFSMS/MVS provides the following programming support for optical disk libraries and drives:

Object Access Method
The Object Access Method lets you create, access, and delete objects on optical, tape, or DASD volumes. You define SMS groups and classes to establish:

- Object storage medium hierarchies and requests for a backup copy of an object
- Whether the object resides on optical, tape, or DASD volume
- When an object should be migrated from one storage medium to another

You can use automatic class selection routines and an installation exit to monitor and control object expiration.

Device Console Services
The Object Access Method and SMS allow you to use the DISPLAY command to show:

- Object Access Method status (the number, type, and availability of defined optical drives in the active configuration)
- A summary of the name, type, and status of Object Access Method control tasks, and information about object processing status
- Details about the number, type, and status of object processing for a specified control task
- Optical and tape library and optical drive status (online, offline, connectivity to the system). The MVS DISPLAY command can be used for the same purpose in the case of tape drives

You can use the VARY command to vary tape and optical libraries, and optical drives on line or off line. The MVS VARY command can be used for the same purpose in the case of tape drives.

Interactive Storage Management Facility
ISMF lets you define optical and tape libraries and optical drives interactively. You do this by specifying the attributes related to the library or drive. This information is stored in an SCDS.

For further information about optical disk libraries and drives, see:

DFSMS/MVS OAM Application Programmer's Reference
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support

For further information about tape libraries and tape drives, see *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Chapter 8. Distributed Data Access with DFSMSdfp

In today's distributed computing environment, applications must often access data residing on different computers in a network. In many cases, the most effective data access services occur when applications can access remote data as if it were local data.

To provide such services, DFSMS/MVS offers Distributed FileManager/MVS, which supports client platforms using APPC communication protocols and DDM file access protocols.

These data access services enable users and applications on heterogeneous client computers in your network to take advantage of storage resources on MVS, including: system-managed storage, high-performance storage access, file access security (RACF), data sharing, and centralized data access.

You should understand the following terms used in this discussion:

client/server *Client* is a computer or process that requests services on the network. *Server* is a computer or process that responds to a request for service from a client.

In the discussion about Distributed FileManager/MVS, the client is also called the *source* and the server is also called the *target*.

user A *user* accesses a service that allows data or other resources to be used. In the discussion of remote data access, *user* is usually an application running on a client (source) system, but it can be a person using a client system as well.

data set/file *Data set* in MVS is generally equivalent to *file* in other environments. In OS/390 UNIX System Services, the OS/390 UNIX file system is a distinct collection of files and directories.

Distributed FileManager/MVS

Distributed FileManager/MVS uses the Distributed Data Management (DDM) protocol, which enables like and unlike computer systems to share file systems across a network. For more information about DDM, See *DDM Architecture: General Information*.

Distributed FileManager/MVS enables your MVS system to act as a server (target) to remote client (source) systems. Distributed FileManager/MVS is designed to work with operating system platforms that support DDM source requests.

With Distributed FileManager/MVS, you can remotely access data on MVS systems from the following IBM systems, or their equivalent:

- Workstations running the OS/2 operating system. Distributed FileManager/MVS on the OS/2 is called DFM/2.
- AS/400 midrange computers running the OS/400 operating system. Distributed FileManager/MVS on the OS/400 is called Source OS/400 DDM.
- System/36 and System/38 mid-range computers.

© Copyright IBM Corp. 1980, 1999

Figure 14 shows the Distributed FileManager/MVS client-server (source-target) relationship.

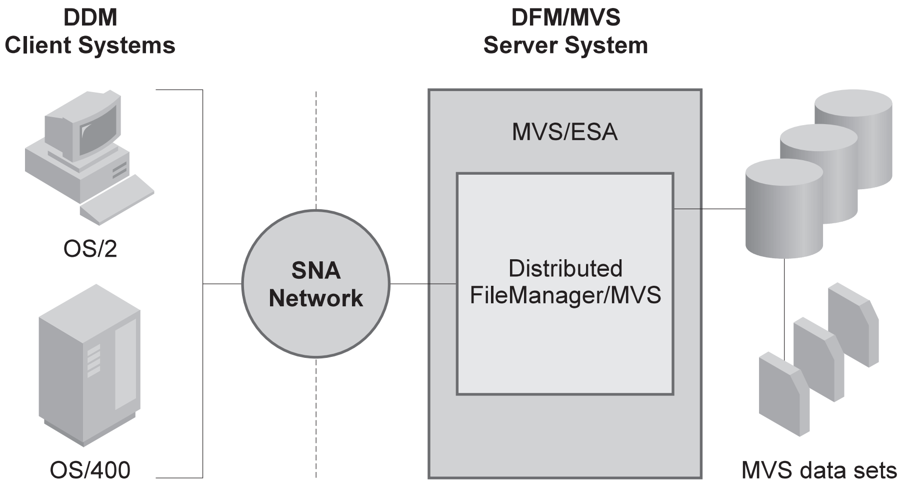


Figure 14. Distributed FileManager/MVS Client-Server (Source-Target) Relationship

Distributed FileManager/MVS offers the following benefits:

- Remote, record-level access to data sets residing on MVS.
- Distributed FileManager/MVS is transparent to the user. Data can be accessed on remote systems as if the data were on a local storage device.
- You can create, update, delete, and rename remotely accessed data sets on MVS.

For more information on Distributed FileManager/MVS, see *DFSMS/MVS DFM/MVS Guide and Reference*.

How Distributed FileManager/MVS Works

Distributed FileManager/MVS uses Distributed Data Management (DDM) and Advanced Program to Program Communication (APPC) protocols to perform remote data access.

DDM a data access architecture. DDM includes a set of rules for accessing data from remote workstations as well as a set of standardized file models and access methods. These rules allow users to access files without concern for where the files are located. Distributed FileManager/MVS uses a subset of DDM commands to respond to remote data access requests.

The DDM architecture is published and therefore is open for non-IBM vendors to use in supporting DDM products of their own. For more information about DDM architecture, See *DDM Architecture: General Information*.

APPC an SNA protocol designed to handle peer-to-peer network conversations between application programs. APPC is also known as LU 6.2. Distributed FileManager/MVS on an MVS target system is set up as an APPC transaction program (TP). During remote data access processing, APPC schedules Distributed FileManager/MVS conversation address spaces on the MVS system. Working with APPC, the Resource Access

Control Facility (RACF) verifies a user's authorization to remotely access MVS data sets.

Figure 15 shows the relationship between an OS/2 Client and an MVS server:

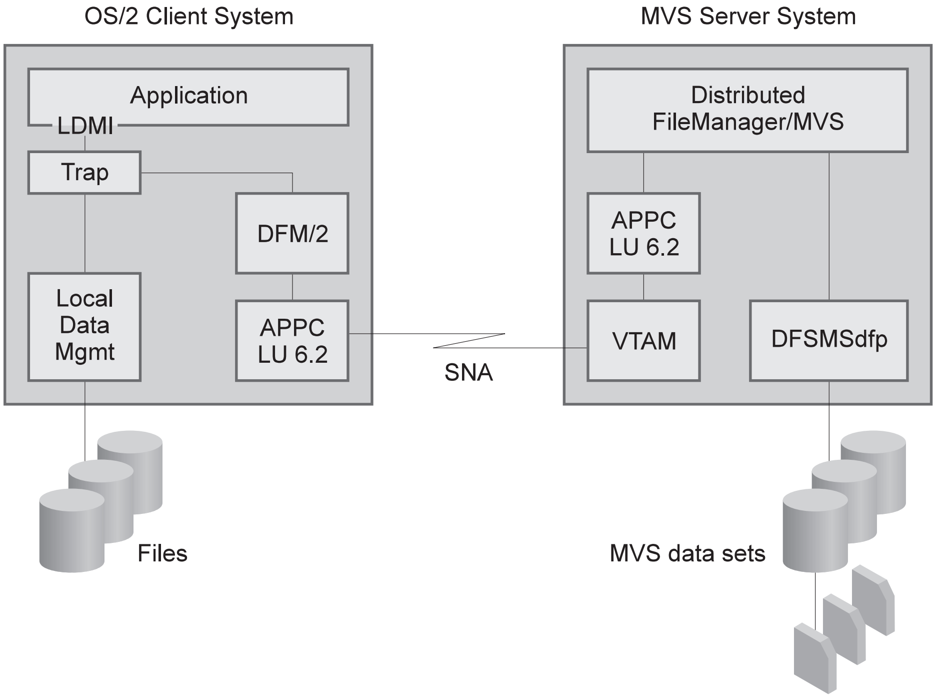


Figure 15. Distributed FileManager/MVS OS/2 Client-MVS Server Flow

As Figure 15 illustrates, an OS/2 application typically accesses data on an MVS target system as follows:

1. An application program running on OS/2 requests data that is remotely located, through a Local Data Management Interface (LDMI).
2. Data access control is passed to DFM/2 (source DDM) which translates the local request into DDM commands and passes them to the OS/2 APPC.
3. OS/2 APPC transmits the request for remote data over a System Network Architecture (SNA) network to the MVS server system.
4. On the MVS system, APPC passes the request to Distributed FileManager/MVS which translates the DDM commands into local data management requests and the requested data is retrieved.
5. The requested data is sent back to the OS/2 application that requested the data.

Accessing MVS Data with Distributed FileManager/MVS

Distributed FileManager/MVS enables users on authorized client systems to remotely access data on MVS server systems as if the data were local to the users. Users can remotely perform such tasks as creating, reading, updating, and deleting MVS data sets using local commands. Because DDM clients and servers communicate through common DDM commands, users on the client systems do not need to use MVS command language.

Distributed FileManager/MVS accepts record access with a record application programming interface (API) or byte stream access with a byte stream API. Distributed FileManager/MVS allows client systems to access the following data set types on MVS server systems:

- Physical sequential data set
- VSAM data set
  - Entry-sequenced data set
  - Key-sequenced data set
  - Relative-record data set
  - Variable-length relative-record data set
- Partitioned data set extended (PDSE) and partitioned data set (PDS) members, treating each member as a separate file

  Note: Distributed FileManager also supports path processing through an alternate index for VSAM data sets.

Reading and Updating Files with Distributed FileManager/MVS

All data sets accessed by Distributed FileManager/MVS must be cataloged in an integrated catalog facility catalog.

You can use Distributed FileManager/MVS with both system-managed and non-system-managed data sets. PDSE data sets, however, must be system-managed in order for you to fully access their members. PDS members are also limited to read access when access is by byte stream. Additionally, the Storage Management Subsystem must be active on the MVS server system for remote data access to occur even if a data set is non-SMS-managed.

Distributed FileManager/MVS ensures data integrity and also provides for simultaneous data sharing of MVS data sets.

Creating Files with Data FileManager

Users on client systems can remotely create any type of data set on MVS server systems to which Distributed FileManager/MVS allows access. Remotely created MVS data sets, however, must be system-managed.

You can remotely create MVS data sets that have associated file attributes. Examples of associated file attributes are file size or file expiration date. Data sets containing file attributes can be examined, moved, copied, and backed-up using MVS utilities.

Using Distributed FileManager/MVS DataAgent

Distributed FileManager/MVS Data Agent is an extension to Distributed FileManager/MVS that allows workstation users of VSAM/x on Windows, OS/2, and RS/6000 to invoke DataAgent routines. Through DataAgent routines, end users can issue TSO commands, CLISTS, or REXX execs to access otherwise unsupported MVS data sets or databases, and gain greater control over processing on the MVS server. For example, a DataAgent routine could be used to extract data from MVS files and databases at the beginning of a VSAM/x application, in preparation for subsequent retrieval by the client through normal VSAM/x interfaces.

DataAgent routines can be user-written, IBM-written, or vendor-written, and are invoked through the VSAM/x component of SMARTdata Utilities (SdU), or from the

OS/2 command line. Several sample routines are provided, showing how to write a DataAgent routine in a high-level language, how to invoke DFSORT, and how to invoke TSO functions such as TSO CLISTS, REXX execs, or TSO commands.

See *DFSMS/MVS DFM/MVS Guide and Reference* for more information.

Chapter 9. Role of the Functional Component DFSMSdss

DFSMSdss is the DFSMS/MVS functional component that copies and moves data to help manage storage, space, and data more efficiently. You can use DFSMSdss to:

Copy and move data
DFSMSdss enables you to copy and move data between like and unlike devices and also copy backup data. For more information, see Copying and Moving Data.

Manage space
DFSMSdss helps you to manage DASD space efficiently. For more information, see “Managing Space” on page 85.

Back up and restore data
DFSMSdss provides host system backup and recovery functions. For more information, see “Backing Up and Restoring Data” on page 86.

DFSMSdss also includes a stand-alone restore program that restores data without a host operating system. For more information, see “DFSMSdss Stand-Alone Services” on page 88.

Convert data
DFSMSdss converts non-system-managed data to system-managed data and system-managed data to non-system-managed data. See “Converting To System-Managed Data” on page 89.

This chapter provides an overview of the DFSMSdss capabilities.

Copying and Moving Data

DFSMSdss is the primary data mover for DFSMS/MVS. A fast, efficient data mover pays for itself when you must move many data sets from old to new DASD. The data movement support that DFSMSdss provides is useful at other times. For example, DFSMSdss allow you to move data sets off a volume for hardware maintenance.

DFSMSdss saves DDM attributes associated with a data set and preserves those attributes during a copy or move operation.

When used with supporting hardware, DFSMSdss also provides concurrent copy capability. Concurrent copy lets you copy or back up data while that data is being used. The user or application program determines when to start processing, and the data is copied as if no updates have occurred.

Copying and Moving Data to Meet Changing Storage Requirements

Moving data is an essential part of storage management. To replace storage devices, add storage capacity, and meet storage requirements, you must move data. DFSMSdss:

- Moves data sets between DASD types and models
- Moves data sets between system-managed and non-system-managed volumes
- Moves data sets off a volume when hardware maintenance is required
- Moves or copies data sets for other purposes

© Copyright IBM Corp. 1980, 1999

DFSMSdss moves data sets from one DASD volume to another volume of the same or different device type (devices are of the same type if both their track capacities and the number of tracks per cylinder are the same). If the device types are the same, DFSMSdss can copy ranges of tracks from a volume or can copy a full volume.

When moving data to a different device type, DFSMSdss offers a significant advantage for a target device having a larger track size than the source device. DFSMSdss is designed to fill the track on the target device as completely as possible (instead of a track-for-track move). Each track can be fully used, holding more records per track on the target than on the source.

You can select data sets to be moved by searching the catalogs and the VTOC.

DFSMSdss allows you to:

- Move an entire VSAM sphere with one copy (that is, the base cluster and all associated alternate index components and paths). You need not move each component individually.
- Copy or move multivolume data sets.
- Retrieve readable data from a damaged volume.
- Delete data sets from the source volume after a successful copy.
- Rename the copied data sets.
- Control data set placement on the target volume by copying to preallocated data sets. In this way, you can place data sets on specific target volumes and even specific tracks.
- Redefine the block size of one or more physical sequential or partitioned data sets (PDS) or partitioned data sets extended (PDSE) while copying those data sets.
- Copy a PDSE to a PDS, or a PDS to a PDSE.
- Copy data sets that have an undefined data set organization (DSORG) to an unlike device of larger capacity. DFSMSdss uses track-for-track replacement to copy or restore an undefined DSORG data set to a target device whose track size is greater than or equal to the source device.
- Move system data sets.
- Specify the target status of system-managed generation data sets.

You can move data sets between system-managed volumes and non-system-managed volumes. Moving or copying invokes the ACS routine to determine the Storage Management Subsystem classes of the target data set. You can specify the storage class name or the management class name, or if you have RACF authorization, you can specify that ACS routines be bypassed. For more information, see “Using SMS Classes and Groups” on page 23 (ACS routines) and “Controlling Access to DFSMSdss Tasks and Data” on page 94 (RACF).

Copying DFSMSdss-Produced Backup Data

If you need extra backup tapes for disaster recovery or for the distribution of dumped data (for example, when generating a new system), you can make from 1 to 255 copies of DFSMSdss-produced backup data.

You can copy a DFSMSdss-backed up sequential data set to a tape or DASD volume. The sequential data set that you are copying can reside on one or more tape or DASD volumes. If the backup data is produced from multiple DASD volumes using a data set dump, you can selectively copy the data from one or more of the DASD volumes.

Managing Space

Allocation algorithms and the frequent creation, extension, and deletion of data sets fragment free space on DASD volumes. The results are an inefficient use of DASD space, an increase in space-related abends (abnormal endings), and degraded performance.

With DFSMSdss, you can effectively manage space by reducing or eliminating free-space fragmentation. DFSMSdss provides features to:

- Compress partitioned data sets
- Release unused space in physical sequential, PDSs, PDSEs, and extended-format VSAM data sets.
- Consolidate free space on DASD volumes.

Compressing Partitioned Data Sets

You can selectively compress partitioned data sets on specified volumes. This procedure removes unused space between members in a partitioned data set. This recovered space is then available for reuse at the end of the data set. This process is not the same as data compression performed in the access methods. You can also use DFSMSdss to release this recovered space.

Releasing Unused Space

DFSMSdss releases allocated but unused space from sequential data sets, partitioned data sets, PDSEs, and extended-format VSAM data sets using either volume or catalog filtering. You can:

- Release space only if the unused space is larger than the number of tracks you specify
- Retain a specified secondary space allocation. This leaves space to add records to the data set after DFSMSdss releases the unused space.
- Release unused space in system data sets

Consolidating DASD Space

DFSMSdss relocates data set extents on a DASD volume to reduce or eliminate free-space fragmentation and provides a report about free space and other volume statistics. You can:

- Find the free space and volume statistics without defragmenting a volume.

- Specify that a volume be defragmented only if the space wasted by fragmentation exceeds a specified threshold.
- Use filtering to exclude selected data sets from defragmentation.
- Lock data sets on shared DASD volumes using dynamic allocation. This prevents other processors in a JES3 environment from accessing the data sets during defragmentation when main device scheduling is active.

When protected data sets are relocated, DFSMSdss erases the data in the old location for security reasons. For more information about this erase-on-scratch feature, see "Resource Access Control Facility Protection" on page 57.

Backing Up and Restoring Data

You can use DFSMSdss to back up data from DASD to tape or other DASD and to restore the backup if the original is lost, damaged, or inadvertently changed. You can also use DFSMSdss to back up application data for disaster recovery and vital records protection.

DFSMSdss saves DDM attributes associated with a data set and preserves those attributes during backup and restore operations. When used with supporting hardware, DFSMSdss also provides concurrent copy capability during backup and sequential data striping capability during backup and restore.

DFSMSdss backs up and restores data sets, specific tracks on a volume, or a full volume.

Backing Up Data

You can select data sets for backup by searching catalogs and the VTOC. You can back up data selectively, using either ISMF or DFSMSdss filtering to choose only the data sets you want. For a description of DFSMSdss filtering, see "Selecting Data Sets by Name or by Characteristic" on page 91.

DFSMSdss allows you to:

- Back up an entire VSAM sphere (that is, the base cluster and all associated alternate index components and paths) by invoking one dump.
- Reset the data-set-changed flags of data sets successfully dumped in a full dump or in a data set dump. The next time you do a data set dump, you can specify that only data sets updated since the last reset are to be dumped.
- Specify whether DFSMSdss continues or cancels a dump when a permanent read error occurs.
- Lock data sets on shared DASD volumes using dynamic allocation. When main device scheduling (MDS) is active, locking data sets prevents other processors in a JES3 environment from accessing those data sets during a dump.
- Dump either all the allocated space or only the used space in physical sequential and partitioned data sets, PDSEs, or data sets with no indicated data set organization (DSORG is undefined). Users typically create the latter with the EXCP macro.
- Dump multivolume data sets.
- Dump and delete system data sets.

- Save space in the dump output data set by compressing the data and optimize the data transfer rate by specifying the number of DASD tracks read at one time.
- Retrieve readable data from a damaged volume.
- Filter on SMS class names. For example, you can filter on a management class to back up only data sets assigned that class.

Restoring Data

You can use DFSMSdss to restore data to DASD volumes from DFSMSdss-produced dump volumes. You can restore data to the same or different device types.

DFSMSdss allows you to:

- Restore an entire VSAM sphere with a single operation.
- Specify whether DFSMSdss continues or cancels a restore when a permanent read error occurs.
- Lock data sets on shared DASD volumes using dynamic allocation. When main device scheduling (MDS) is active, locking data sets prevents other processors in a JES3 environment from accessing those data sets during a restore.
- Rename a data set during a restore. You can create a new data set with a new name instead of replacing the original data set on a DASD volume. You can change either the entire name or part of the name.
- Redefine the block size of physical sequential or partitioned data sets or PDSEs while recovering those data sets.
- Restore data sets that have an undefined DSORG to an unlike device of larger capacity.
- Specify the target status of system-managed generation data sets.
- Restore a user catalog without bringing down your application program. This allows continuous applications, such as DB2 and IMS/ESA, to remain active while a damaged user catalog is being replaced.
- Restore data sets to system-managed volumes and non-system-managed volumes. If you have RACF authorization, you can specify that the ACS routines be bypassed. Otherwise the ACS routines are invoked to assign SMS classes to the data sets.

Snapshot Support (DFSMSdss)

Snapshot is a function of the RAMAC Virtual Array (RVA) that allows you to make a very quick copy of a set of tracks (an entire volume, a data set, or just a random set of tracks). The copy operation is completed with only a few I/Os to the device.

DFSMSdss provides two types of Snapshot support

"Native Snapshot" Support

Data is "snapped" (quickly copied) directly from the source location to the target location. This function occurs when you issue a DFSMSdss COPY command to copy volumes, tracks, or data sets from one DASD volume to another. DFSMSdss uses this method whenever the source and target data are on like devices in the same partition on the same RVA subsystem, and no reblocking is required. DFSMSdss can use this method whether or not the CONCURRENT keyword is specified. With "native Snapshot", the copy of the data is logically and physically complete as soon as the "snap" is complete.

"CC-compatible Snapshot" Support
Data is "snapped" from the source location to an intermediate location and then gradually copied to the target location using normal I/O methods. This function occurs when you issue either the DFSMSdss COPY or DUMP command and specify the CONCURRENT keyword. As the name implies, this method operates in a fashion almost identical to existing Concurrent Copy (CC) support. All DFSMSdss users and callers of the DFSMSdss API (such as DFSMShsm, DB2, and IMS) can continue to use the CONCURRENT keyword and receive functionally identical CC support, and on a wider range of devices. Using "CC-compatible" Snapshot, the copy or dump of the data is logically complete after the source data is "snapped" to the intermediate location, and then physically complete after the data has been moved to the target media. It is also possible to perform Concurrent Copy on VM minivolumes using "CC-compatible" Snapshot.

DFSMSdss Stand-Alone Services

The new Stand-Alone Services function is available to all supported releases of DFDSS and DFSMSdss. The Stand-Alone Services function completely replaces and makes obsolete the previous Stand-Alone Restore function, which is no longer available as part of DFSMSdss.

This new version of IBM’s Stand-Alone restore function is a single-purpose program designed to allow the system programmer to restore vital system packs during disaster recovery without needing to rely on an MVS environment. Stand-Alone Services runs independently of a system environment either as a “true” stand-alone system or under a VM system.

Stand-Alone Services can perform either a full-volume or a tracks restore from dump tapes produced by DFSMSdss or DFDSS, and offers the following benefits when compared to the previous DFSMSdss Stand-Alone functions:

- Provides user-friendly commands to replace the previous control statements
- Supports IBM 3494 and 3495 Tape Libraries, and 3590 Tape Subsystems
- Supports IPL-ing from a DASD volume, in addition to tape and card readers
- Allows you to predefine the operator console to be used during Stand-Alone Services processing

The Stand-Alone Services program operates on an IBM System/370 processor in either ESA/370 mode, 370-XA mode, or System/370 (S/370) mode. It also runs on an IBM System/390 processor in ESA/390 mode or S/370 mode. The Stand-Alone Services program can run on a machine that is in BASIC or LPAR mode, or in a virtual machine under VM.

Refer to *DFSMS/MVS DFSMSdss Storage Administration Reference* for a complete description of the DFSMSdss Stand-Alone Services function, including a listing of supported devices and instructions for creating the IPL-able Stand-Alone Services core image. Messages for the Stand-Alone Services function are found in *OS/390 MVS System Messages, Vol 1 (ABA-ASA)*.

Converting To System-Managed Data

DFSMSdss is the primary tool for converting system-managed data to non-system-managed data and non-system-managed data to system-managed data. DFSMSdss supports conversion with or without data movement.

Converting by Moving Data

DFSMSdss commands can move and convert data sets between system-managed and non-system-managed volumes. DFSMSdss supports data conversion by:

- Recognizing and preserving SMS classes. Each system-managed data set has a set of associated class names. These names can identify the data set's data class, management class, and storage class. DFSMSdss recognizes and, if specified, preserves these names when copying, dumping, and restoring data.
- Copying, dumping, and restoring system-managed data sets. DFSMSdss moves data sets between system-managed volumes and non-system-managed volumes. The SMS class names are either added to or taken away from the data set.
- Filtering by SMS class names. You can filter both system-managed and non-system-managed data sets by class names.

Converting Volumes without Moving Data

DFSMSdss provides a command for converting volumes without data movement. Before attempting to convert volumes, you can verify that DFSMSdss can convert all data sets on that volume.

Chapter 10. DFSMSdss Facilities

This chapter describes facilities that make DFSMSdss a flexible and powerful tool. DFSMSdss lets you:

- Choose between logical and physical processing
- Select data sets by filtering on criteria you specify
- Invoke DFSMSdss from a variety of sources
- Control run-time options using auxiliary commands
- Print DASD data.

Choosing between Two Processing Methods—Logical and Physical

DFSMSdss can perform either logical or physical processing. If you dump a data set logically, DFSMSdss restores it logically; if you dump it physically, DFSMSdss restores it physically. Logical processing operates against data sets independently of physical device format. Physical processing moves data at the track-image level and operates against volumes, tracks, and data sets. Each type of processing offers different capabilities and advantages.

Logical Processing

During logical processing, DFSMSdss treats each data set and its associated information as a logical entity. DFSMSdss processes an entire data set before beginning the next one. DFSMSdss moves each data set as a set of data records, allowing data movement between devices with different track and cylinder configurations. DFSMSdss uses the catalog and the VTOCs to select data sets for logical processing.

Physical Processing

During physical processing, DFSMSdss moves data based on physical track images. Because DFSMSdss moves data at the track level, the target device's track sizes must equal the source device's track sizes. Therefore, you can use physical processing only between like devices. Physical processing operates on volumes, ranges of tracks, or data sets. For data sets, physical processing relies only on volume information in the VTOC and VVDS for data set selection, and processes only that part of a data set residing on the specified input volume.

Selecting Data Sets by Name or by Characteristic

You can select data sets for DFSMSdss processing by filtering on criteria you specify. DFSMSdss can filter on fully or partially qualified data set names and on various data set characteristics.

Filtering by Data Set Names

You can select data sets for inclusion in or exclusion from DFSMSdss processing based on data set name. Data set names for DFSMSdss filtering can be fully qualified or partially qualified and can consist of one or more qualifiers.

© Copyright IBM Corp. 1980, 1999

Filtering by Data Set Characteristics

Besides filtering on data set names to process groups of data sets, you can filter on data set characteristics, such as:

- Allocation type (cylinder, track, block, absolute track, or movable)
- Creation date (absolute or relative)
- Expiration date (absolute or relative)
- Last-referenced date (absolute or relative)
- Data set organization
- Data set size (number of allocated or used tracks)
- Number of extents
- Whether the data set is single-volume or multivolume
- Whether the data-set-changed flag is on or off
- SMS class names
- Whether a data set is cataloged using the standard catalog search order.

Invoking DFSMSdss

You can invoke DFSMSdss processing through job control language (JCL), ISMF, or an application program.

Invoking DFSMSdss with JCL

You can use JCL statements to invoke DFSMSdss and to define the data sets that DFSMSdss needs.

Invoking DFSMSdss with ISMF

You can use ISMF's menu-driven panels to submit commands to DFSMSdss. Simply fill in the panels with the values you want, and ISMF generates the JCL.

For more information about ISMF, see “Interactive Storage Management Facility and SMS-Managed Storage” on page 37. For more information on using DFSMSdss with ISMF, see the ISMF online help panels and *DFSMS/MVS Using ISMF*.

Invoking DFSMSdss from Your Application Program

You can invoke DFSMSdss from an application program using the application interface. You can also invoke DFSMSdss to run in a separate address space by using the cross-memory application interface. These interfaces allow you, for example, to specify control variables, and to gather information on free and used space on a volume or in a data set. Any program that calls DFSMSdss must either be authorized by APF or must invoke DFSMSdss using the cross-memory application interface. For more information on APF protection and authorization, see “Authorized Program Facility (APF) Authorization” on page 58.

When you invoke DFSMSdss from an application program, you can provide a user interaction module (UIM) to interact with DFSMSdss during processing at various points. The UIM lets you:

- Replace, insert, delete, or modify a SYSIN record after DFSMSdss reads it or a SYSPRINT record when DFSMSdss is ready to print it

- Replace, insert, delete, or modify a write-to-operator (WTO) message before DFSMSdss writes it

- Control processing and gather auditing information for an individual data set during a logical copy, dump, or restore
- Control processing for concurrent copy operations
- Insert statistical records during a logical dump
- Intercept records being dumped or supply records to be restored

Using Auxiliary Commands to Control Run-Time Options

Auxiliary commands can control the way DFSMSdss runs. For example, you can use auxiliary commands to:

- Start serial task scheduling (the default). DFSMSdss executes only one task at a time.
- Start parallel task scheduling. DFSMSdss executes two or more tasks concurrently if the required system resources (such as virtual storage, DASD, or tape volumes) are available.
- Control how DFSMSdss runs based on the return codes of completed operations.
- Write a message to the system console.
- End your DFSMSdss job after the currently running operations and scheduled tasks complete.

Printing DASD Data

You can use DFSMSdss to print DASD data to the SYSPRINT data set or to a sequential data set in print format. The records in the output data set are blocked to better use DASD space. For data set printing, tracks are printed in the logical sequence of the data set on the volume, not in the physical cylinder and head sequence.

You can print the following:

- A non-VSAM data set specified by a fully qualified name. You need not identify the data set location.
- VSAM data sets, including key-range data sets at the component level but not at the cluster level.
- Ranges of tracks.
- All or part of the VTOC. You need not identify the VTOC location.

You can optionally print only the tracks on which read errors occur for any of the preceding items.

When an error occurs while DFSMSdss is trying to read a record, DFSMSdss tries to print the record that is in error.

Controlling Access to DFSMSdss Tasks and Data

You can use RACF to protect resources such as DASD volumes, tape volumes, and data sets against unauthorized access. You can also use RACF to limit the use of certain DFSMSdss functions to privileged users.

Protecting Data

You can grant access to a data set at the volume level or at the data set level. DFSMSdss first invokes RACF to check the user's access authority to the volume. If the user has the required RACF authority at the volume level, DFSMSdss does not check at the data set level. RACF checks at the data set level only if:

- The user does not have the required volume level authority.
- The volume is not protected.
- DFSMSdss cannot determine the protection status of the volume.

A data set can be either RACF-protected, password-protected, or both. When the data set is protected by RACF, DFSMSdss ignores the data set password.

When copying or restoring data, DFSMSdss checks to make sure that users have sufficient authority to create or overlay the target data sets. In addition, DFSMSdss tries to protect the target data sets with the same RACF protection as the source data sets.

DFSMSdss supports the DASD erase-on-scratch attributes defined in the RACF profile of the data set. For more information, see “Resource Access Control Facility Protection” on page 57.

Protecting DFSMSdss Commands and Keywords

By defining the appropriate RACF FACILITY class profile you can protect certain DFSMSdss commands and keywords against unauthorized use. DFSMSdss invokes RACF checking to ensure that users have sufficient authority to perform the function. For more information on the FACILITY class profiles used by DFSMSdss, see *DFSMS/MVS DFSMSdfp Storage Administration Reference*.

Customizing DFSMSdss with Installation Exits

Installation exits provide a way to tailor DFSMSdss during installation. Replace these exit routines with your own routines to:

- Control or override the authorization checking of protected data sets
- Control the duration of the enqueue of the VTOC
- Verify or change the block size of a reblockable data set
- Specify installation options and defaults.

For more information on DFSMSdss installation exits, see *DFSMS/MVS Installation Exits*.

Chapter 11. Role of the Functional Component DFSMShsm

DFSMShsm is the DFSMS/MVS functional component that provides automatic management of low-activity and inactive data, and automatic backup and recovery of active data, in both system-managed and non-system-managed environments. DFSMShsm accomplishes this by providing:

Automatic Storage Management
DFSMShsm is a tool that improves productivity by effectively managing storage. DFSMShsm uses a hierarchy of storage devices in its automatic management of data, relieving professional personnel and end-users from manual storage management tasks, and improving DASD utilization. For more information, see "Device Hierarchy" on page 96.

Space Management
DFSMShsm automatically manages DASD space by enabling active data sets to be kept on fast-access storage devices. DFSMShsm frees available space on user volumes by releasing over-allocated space, deleting eligible data sets, and by moving low-activity data sets to lower-cost-per-byte devices, ensuring that your most expensive storage contains only data that is being used frequently.

By using DFSMShsm in conjunction with your ACS routines to implement the tape mount management methodology, you can write multiple output data sets to a single tape, with a single tape mount. This not only improves tape use, but also greatly reduces the number of tape mounts required by your installation.

For more information, see "Managing Space" on page 98.

Availability Management
DFSMShsm makes data available by automatically copying new and changed data sets to backup volumes. If the original data sets are damaged or accidentally deleted, having backup versions ensures that the data can be made available. DFSMShsm also provides the ability to back up aggregate groups of critical data sets and programs that can be taken to remote locations and used for disaster recovery or other business needs. For more information, see "Managing Availability" on page 106.

DFSMShsm also provides the secondary host promotion function to improve data availability in instances where a DFSMShsm host processor experiences a failure in a multiprocessor environment. Using the cross-system coupling facility (XCF) in either a basic or parallel sysplex environment, the secondary host promotion function enables a secondary host to take over the unique functions that were performed by a failed primary host. The secondary host promotion function also enables other DFSMShsm hosts to take over secondary space management functions from either a primary or secondary host that has failed. If a promoted host also fails, any remaining host that is eligible for promotion will take over.

DFSMShsm provides a number of functions to support space and availability management. Full exploitation of DFSMShsm services in DFSMS environments requires the use of DFSMSdss for certain functions.

This chapter introduces DFSMShsm. It discusses how DFSMShsm can automatically manage low-activity and inactive data and backup and recover active data.

© Copyright IBM Corp. 1980, 1999

Device Hierarchy

Device hierarchy is defined as levels of storage devices with each level having its different access speeds, costs per byte, and storage capacities.

At the top of the device hierarchy is the DASD storage control with its optional, high-speed, cache storage.

In the middle of the device hierarchy is DASD. Data that needs to be frequently accessed is typically stored on DASD.

The lowest level of the device hierarchy is removable media. Removable media can be used with optical and tape library dataservers, for the fastest response time in this level of the hierarchy, as well as with standalone optical and tape drives with shelf storage. Removable media has the slowest initial access time because the data is stored offline; however, it has the lowest cost per megabyte when used efficiently. The storage capacity of removable media is limited only by the number of volumes your installation is willing to use and manage. In addition, with removable media, volumes can be removed to a remote site to protect them from damage in the event of a disaster.

Data Hierarchy

DFSMShsm supports advanced concepts in data hierarchy not always found in similar products. The data hierarchy is comprised of:

Active data Active data is data that is frequently used, and seldom migrated. The data normally resides on DASD, for fast access, and is in user format. The DASD containing active data are called level 0 volumes, and they can be managed by DFSMShsm.

Low activity data Low activity data is data that is sometimes used, but not often. This type of data is eligible to be migrated by DFSMShsm, or has already been migrated by DFSMShsm, to a more cost-efficient type of storage.

Inactive data Inactive data is data that has been dumped or is at the incremental backup level. Dumps and incremental backups are taken of the active data and of the DFSMShsm control data sets and journal. These dumps and incremental backups can then be used to recover data sets, or entire volumes, if they are lost due to damage or accidental deletion.

Inactive data also includes backups of aggregate groups. These backups can be transported to a remote site for disaster recovery, or they can be used for other business purposes.

DFSMShsm Control Data Sets

DFSMShsm control data sets are system-type data sets that DFSMShsm uses to keep track of all DFSMShsm-owned data. They consist of migration, backup, and offline control data sets. The control data sets are an inventory of low-activity and inactive data that has been stored by DFSMShsm, and they are used by DFSMShsm to manage its environment. DFSMShsm logs its transactions and maintains multiple backup versions of its CDSs for recovery purposes as specified by the user. CDSs can be larger than 4 gigabytes in size.

The control data sets can be accessed in record-level sharing (RLS) mode. VSAM KSDS extended addressability (EA) format is supported for CDSs when accessing the CDSs using RLS. Accessing the control data sets in RLS mode allows DFSMShsm to exploit parallel sysplex technology and to benefit from the serialization features of the coupling facility when performing functions on two or more processors.

When updating the CDSs in a sysplex environment with more than one HSMplex, DFSMShsm offers GRSplex serialization to avoid contention between processing in each HSMplex. Other functions, such as level 1 to level 2 migration, can benefit by avoiding contention when running concurrently in two or more separate HSMplex environments in the same GRSplex.

DFSMShsm-Owned Volumes

DFSMShsm-owned volumes contain low-activity and inactive user data sets that have been moved (migrated data) or copied (dumped and backed up data) into the DFSMShsm-owned portion of the storage hierarchy. The data is stored in a space-saving format. The data stored in a space saving-format is not directly accessible by users, but must be returned to a level—0 volume in user format before users can access it. Information about DFSMShsm-owned data is recorded in the inventory maintained in the DFSMShsm control data sets.

Low-activity data is stored on migration level 1 (ML1) and migration level 2 (ML2) volumes. The DFSMShsm automatic management of ML1 and ML2 volumes can be supplemented by DFSMShsm commands to manually manage them.

Backup tapes contain the backup copies of data sets placed there on a daily basis. Spill backup tapes contain data moved from backup volumes as a result of recycle processing. The data is written in a DFSMShsm space-saving format.

The recycle function is a tape consolidation process performed on ML2 and backup tapes that become sparsely populated with valid data.

Dump tapes contain all the data of a physical volume and are written in a DFSMSdss space-saving format.

Several dumps from different DASD volumes can be automatically stacked onto a single tape.

Aggregate data tapes contain user-defined groups of data sets that have been copied by aggregate backup for recovery at a remote site.

Aggregate control tapes can contain control information, data from L0 DASD, data from ML2 and user tape, and instruction and log data.

ML2 and backup tapes can have alternate tapes. The alternate tapes are produced by either duplicating the original tapes, or by using the duplex tape function to concurrently create a copy when the original ML2 or backup tapes are created. Alternate tapes provide site disaster protection, supplementing the ABARS (aggregate backup and recovery support) aggregates and protecting against media damage.

Managing Space

You can use DFSMShsm to keep DASD space available for users in order to meet the service-level objectives for your system. DFSMShsm can automatically and periodically move low-activity data sets from user DASD volumes to DFSMShsm-owned volumes. DFSMShsm also reduces the space occupied by data on both the user DASD volumes and the DFSMShsm-owned volumes.

To provide you with more control, daily space management is divided into automatic primary and automatic secondary space management functions. Each of these functions can have a processing cycle. Each can be started at a specified time of day and end when all the data managed by DFSMShsm has been processed or when the ending time of day, as specified by the storage administrator, is reached.

Automatic Primary Space Management

During automatic primary space management, DFSMShsm can process a maximum of 15 volume migration tasks concurrently. This activity consists of deletion of temporary data sets, deletion of expired data sets, release of unused overallocated space, and migration. Each task processes its own separate user DASD volume. The storage administrator selects the maximum number of tasks that can run simultaneously, as well as specifying which days and the time of day the tasks are to be performed.

Figure 16 on page 99 shows the migration paths of data sets managed in system-managed and non-system environments.

Processing System-managed Volumes

If a volume is system-managed, DFSMShsm uses the storage group’s threshold of occupancy as a measure of how much free space to provide. The volumes processed are those in storage groups having the automatic migration attribute. Storage groups can also be used to restrict the processing to a single system or a system group.

DFSMShsm manages each data set on a volume according to the management class attributes associated with the data set. As the different data sets are encountered, DFSMShsm performs the following operations:

- Deletes temporary data sets that are unintentionally left at the end of the job.
- Deletes data sets that have an explicit or implied expiration date. If the data set has no explicit expiration date, it is controlled by the management class attributes. The storage administrator can control whether DFSMShsm deletes expired data sets having explicit expiration dates.

- Releases overallocated space as determined by the management class option for system-managed data sets.
- Migrates system-managed data sets from DFSMShsm-managed level 0 volumes to ML1 or ML2 volumes. Only those data sets that have not been referred to for at least as many days as is specified in the management class are migrated, and only until the low threshold of occupancy is reached for the volume. However, generation data groups can be given criteria for migration based on their relative generation number rather than on their inactive age. Reduces the extents of all eligible data sets during migration according to parameters specified by the storage administrator.

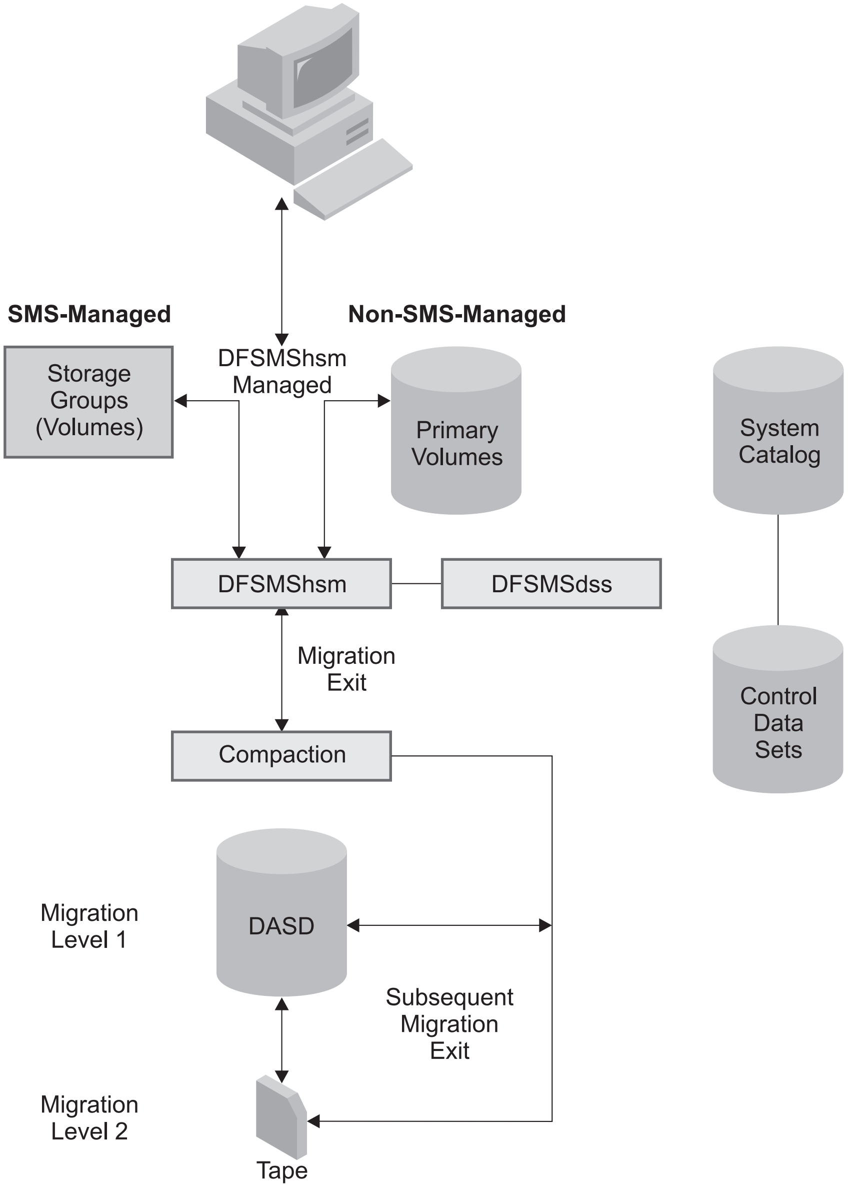


Figure 16. Automatic Volume and Secondary Space Management

Processing Non-System-managed Volumes

Figure 16 on page 99 also shows automatic volume and secondary space management:

If a volume is non-system-managed, DFSMShsm manages each data set on a volume according to the attributes of the volume on which the data set resides. As each volume is encountered, DFSMShsm deletes:

- Temporary utility and list data sets to free space. The storage administrator can specify the parameter that controls deletion of the list and utility data sets.
- Expired data sets that have an explicit expiration date. The storage administrator can control whether DFSMShsm deletes expired data sets having explicit expiration dates.

The kind of volume processing used is specified by DFSMShsm parameters for each volume. In addition, DFSMShsm does *only* one of the following as each non-system-managed volume is encountered:

- Deletes all eligible data sets that have reached a specified inactive age and that are not protected by an unexpired expiration date.
- Retires all eligible data sets that have reached their specified inactive age and that have a current backup version. The current backup version is then flagged to prevent inadvertent deletion of the data set.
- Migrates eligible data sets to ML1 or ML2 volumes. Two kinds of volume processing are supported during migration: level-of-occupancy threshold and age, or age alone.
- Reduces the extents of all eligible data sets during migration, according to parameters specified by the storage administrator.

Automatic Secondary Space Management

Automatic secondary space management prepares the computing system for the automatic primary space management work load by freeing space on DFSMShsm-owned volumes. Secondary space management deletes expired migrated data sets and migrates eligible data sets from ML1 volumes to ML2 volumes. The storage administrator specifies the day in the cycle and the time of day automatic secondary space management is to be performed.

Automatic Interval Migration

Automatic interval migration is an option that invokes migration when DFSMShsm-managed volumes become full during high-activity periods. If the storage administrator chooses this option, DFSMShsm automatically checks the level-of-occupancy of all DFSMShsm-managed volumes periodically. If any volume’s level of occupancy exceeds a given threshold, DFSMShsm automatically performs a subset of the space management functions on the volume. The threshold you select should be one that would be exceeded only when your installation’s activity exceeds its usual peak. For those volumes requiring interval migration, DFSMShsm can process up to 15 volume migration tasks concurrently.

During automatic interval migration on a volume, the expired data sets are deleted, then the largest eligible data sets are moved first so that the level of occupancy threshold can be reached sooner. Data sets are not migrated from ML1 to ML2 volumes during interval migration.

For *system-managed volumes*, DFSMShsm uses the level-of-occupancy thresholds established for the storage group to perform this function.

For *non-system-managed volumes*, DFSMShsm uses the level-of-occupancy thresholds established for the DFSMShsm-managed volume to perform this function.

Automatic Recall

The recall process returns a *migrated* data set from a ML1 or ML2 volume to a DFSMShsm-managed volume. When a user refers to the data set, DFSMShsm reads the system catalog for the volume serial number. If the volume serial number is MIGRAT, DFSMShsm finds the migrated data set, recalls it to a DFSMShsm-managed volume, and updates the catalog. The result of the recall process is a data set that resides on a user volume in a user-readable format. The recall can also be requested by a DFSMShsm command.

Figure 17 on page 102 shows the recall paths of data sets managed with and without system-managed storage.

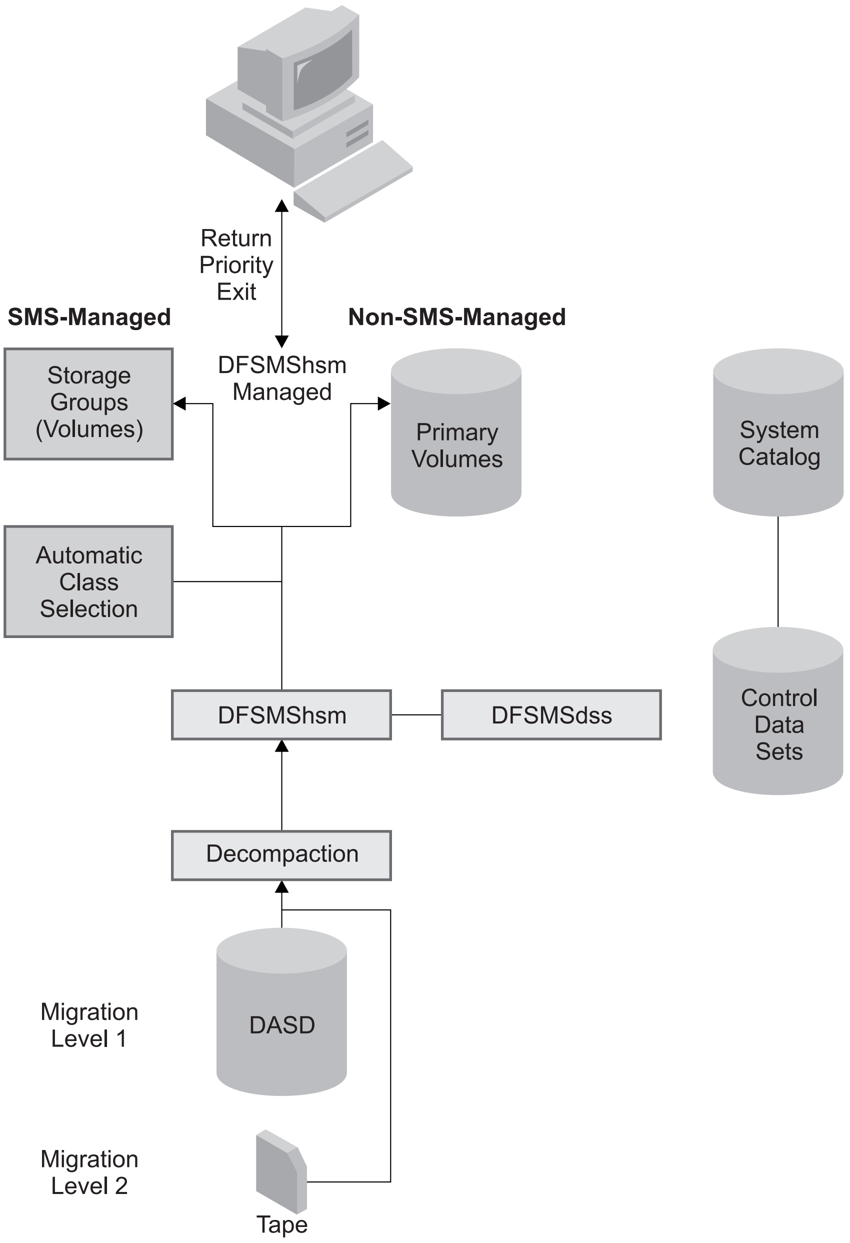


Figure 17. Automatic Recall

For *system-managed environments*, DFSMShsm invokes the automatic class selection routines to determine whether a data set should be recalled as managed with or without system-managed storage. Automatic class selection chooses the storage group to which the data set is recalled. The system chooses from the storage groups the specific volumes where the data set will reside.

For *non-system-managed environments*, DFSMShsm returns a migrated data set to the DFSMShsm-managed volume with the most free space or to a specified target volume.

Saving Space on DFSMShsm-owned Volumes

DFSMShsm can save space on migration volumes with its space-saving functions. DFSMShsm generally moves and copies only used, not over-allocated, space to its migration volumes. The space-saving functions are:

- Data compaction
- Small data set packing
- Partitioned data set compression
- Reblocking of user data sets
- Optimum blocking of data on owned DASD
- Tape recycle

These DFSMShsm functions are available in system-managed and non-system-managed environments.

The combined effects of these space-saving functions can be significant. For example, small data set packing with data compaction can allow up to 24 small data sets to occupy a single DASD track on ML1, whereas under normal conditions, a DASD track can contain no more than one data set. Compaction and optimum DASD blocking can result in additional space savings for data sets too large for small data set packing.

Data Compaction

DFSMShsm can save space on DFSMShsm-owned volumes by compacting each data set as it is processed. The data compaction functions are available with or without using the DFSMSdss component of DFSMS/MVS as the data set mover. DFSMShsm can compact data during data set backup and migration, or by using DFSMSdss compression during physical volume dumps and aggregate backup.

You can also utilize the compaction algorithms of tape devices, such as the IBM 3480, 3490, 3490E, and 3590-1 to compact tape data.

The IBM RAMAC Virtual Array Subsystem also provides its own data compaction, making compaction by either DFSMShsm or DFSMSdss unnecessary.

Data compaction is transparent to users and application programs and occurs during automatic or command processing of data sets or volumes. DFSMShsm automatically decompacts data sets during recall operations, even if your installation later decides to disable data compaction.

Note: The compaction provided by the improved data recording capability can be used in conjunction with sequential access method (SAM) compression, virtual storage access method (VSAM) compression, and DFSMShsm software compaction. However, DFSMShsm software compaction is bypassed if a data set uses SAM or VSAM compression.

Small Data Set Packing

Small data set packing (SDSP) saves space on ML1 volumes by writing small data sets (fewer than 800 KB) as records in a VSAM data set called an SDSP data set. The space saving occurs because the DASD data set allocation requires allocation in full tracks. SDSP lets you give those data sets allocation in 2093-byte units.

Storage administrators can specify the size of data sets eligible to migrate to SDSP data sets either in kilobytes or tracks. The storage administrator must specify which ML1 volumes have SDSP data sets.

For system-managed data sets managed, management class attributes can keep the data set on ML1. For non-system-managed data sets, DFSMShsm provides a subsequent migration exit that can be used to keep small data sets in SDSP data sets.

Partitioned Data Set Compression

Partitioned data set compression reduces the allocated space for a data set as members are deleted or replaced. Users sometimes hesitate to compress their partitioned data sets because the operation is annoying and time-consuming. DFSMShsm provides a safe and automatic method of compressing partitioned data sets during space management by doing migration and recall when the partitioned data set exceeds a certain number of extents.

DFSMShsm maintains user information that is in partitioned data set directories when it performs partitioned data set compression.

User Data Set Reblocking

During recall and recovery, the process of reblocking user data sets changes the number of records in a physical block and thereby uses the space on the DASD user volume more efficiently. Data movement using DFSMShsm reblocks physical sequential data sets. Data movement using DFSMSdss reblocks physical sequential and partitioned data sets.

The storage administrator can control data set reblocking during recall and recovery by using the data set reblocking exit.

Optimum Blocking of Data on DFSMShsm-Owned DASD

Optimum blocking of data is an option that saves space on DFSMShsm-owned DASD volumes by using an optimum block size for storing the maximum data on each track. The optimum block size is determined based on the device type of the DASD device.

Tape Recycle

Data on tapes is invalidated over time by the expiration or recall of migrated data sets or the generation of more recent backup data sets. DFSMShsm provides the capability of recycling backup or migration tapes when they contain less than an installation-defined percentage of valid data. Recycling transfers the valid data sets from these tapes and consolidates the data onto fewer tapes, thus leaving the recycled tapes available for reuse. Tape recycle can also be used to move data from one tape technology type to another. Options are available to select input tapes by volume serial ranges, which can be used to identify tapes of a given technology.

Tapes eligible for recycle processing are logically sorted, and those containing the least amount of valid data (per cartridge type, standard or enhanced capacity) are processed first. This provides the highest number of tapes being returned for reuse early in the recycling process.

To facilitate rapid recycle processing, up to 15 tape-processing tasks can run simultaneously on a processor, with multiple tape buffers being used for improved throughput. The number of recycle tape-processing tasks can be changed dynamically, even during recycle processing, to any number from one to 15. This lets you free tape drives for uses other than recycle processing, should the need arise.

For efficiency:

- An input tape drive allocated for the recycle of a tape, or of a connected set of tapes, remains allocated until all input tapes have been processed.
- Recycle processing can be set to automatically quiesce when a predefined net number of tapes are returned to scratch.
- The RECYCLE TAPELIST command can generate a list showing which tapes will be selected for recycle processing. For customers that have manual tape operations, the TAPELIST option on recycle will generate a pull list in groups that are sorted in alphanumerical order, and a mount list in groups that are sorted by amount of valid data.

Dump tapes and ABARS tapes can be automatically reused without recycle; dump tapes when they reach the end of their retention period, ABARS tapes when they reach an expiration date, or when aggregate roll-off occurs.

If data in single-file format on a backup or ML2 tape is overwritten or becomes unreadable, DFSMShsm can retrieve all the undamaged data from that tape.

Managing Space Manually

The direction of space management is toward fully automatic operations. Occasionally, during transition to a system-managed environment or until all data can be managed consistently, certain operations must be performed outside the policies and parameters that establish an automatic operation.

For example, a level—0 volume not yet managed by DFSMShsm might be out of space, or a user might want to explicitely delete a backup version that is no longer needed. To provide for these conditions, DFSMShsm has a set of commands that can be used to perform space management functions on both individual data sets and volumes. There are also commands to query DFSMShsm and to perform functions that are infrequent or have not been made completely automatic. DFSMShsm provides commands intended for use by storage administrators, operators, and end-users. For more information, see “Protecting DFSMShsm Commands and Parameters” on page 121.

Adding New DASD to a System

DFSMShsm provides a command to allow the storage administrator to move all the data off a DASD migration volume to other migration volumes. This command makes it practical to clear a DASD migration volume so that it can be replaced with a different type of storage device. You can also use this command to empty a DASD backup volume.

For information about converting your level 0 volumes to new DASD, see “Copying and Moving Data to Meet Changing Storage Requirements” on page 83.

Managing Availability

Availability management must ensure that a current backup copy of a data set, a volume, or a set of application data sets is available for recovery at a point in time. To accomplish this objective effectively requires that planning be done from the viewpoint of recovery. What is needed to recover from a logical error or physical loss of a data set? Of a volume? Of all the data sets associated with critical applications? Of all the data sets at a particular computing facility?

DFSMShsm availability management provides for recovery of:

- Critical user applications at remote sites in the event of a physical loss of the computing facility
- User data sets and volumes at the local site in the event of a logical or physical loss
- DFSMShsm control data sets and journal

To protect your investment in application programs and data, use aggregate processing as your disaster survival solution. Rather than managing disaster survival by dumping and restoring volumes, aggregate processing manages disaster survival by copying the data sets belonging to an application, along with relevant control information, on a set of portable device-independent tape data and control files. Thus, critical applications can be recovered at one or more remote sites.

To protect your investment in people and increase the productivity of personnel, use DFSMShsm to invoke and manage DFSMSdss physical volume dump in conjunction with DFSMShsm incremental backup functions. This combination reduces the frequency of volume dumps but maintains its currency. See “User Data Sets and Volumes” on page 108.

Critical User Applications

You can use DFSMShsm to backup and recover your critical applications. If more than one remote site is available for recovery, then concurrent recovery might be possible. The planning process must review the recovery process to determine which data sets should be defined in each aggregate to be managed as a single entity.

Backing Up Aggregate Groups

Aggregate backup and recovery is a process to back up and recover any user-defined group of data sets that are vital to your business. Up to 64 ABARS commands can run concurrently. Users can automate the aggregate backup process by having a DFSMShsm assembler macro submit the ABACKUP command. For more information about aggregate groups, see “Defining Aggregate Groups for Disaster Backup and Recovery” on page 32.

These data sets are identified as an aggregate group defined with the DFSMS/MVS ISMF panels. DFSMShsm uses the data set lists, along with other control information stored in the associated aggregate group and management class, to manage the aggregate backup process.

DFSMShsm works with the other functional components of DFSMS/MVS to provide enhanced backup and recovery capabilities for aggregate groups. When you define the aggregate group, you specify characteristics of that group in its management class. You can specify, for example, how many copies of each backup version you want. The backup copies are managed according to the aggregate group attributes you define in the management class. You can use the same management class attributes for multiple aggregate groups whose backup copies have the same management needs. The attributes assigned to an aggregate group are tracked by the aggregate backup and recovery function and can be used to create the proper environment at the recovery site.

The DFSMS/MVS functional component DFSMSrmm can be used to track the movement of the aggregate backups, further automating your disaster backup process. See Chapter 13, “Role of the Functional Component DFSMSrmm” on page 123.

Figure 18 shows the movement of data sets during aggregate backup.

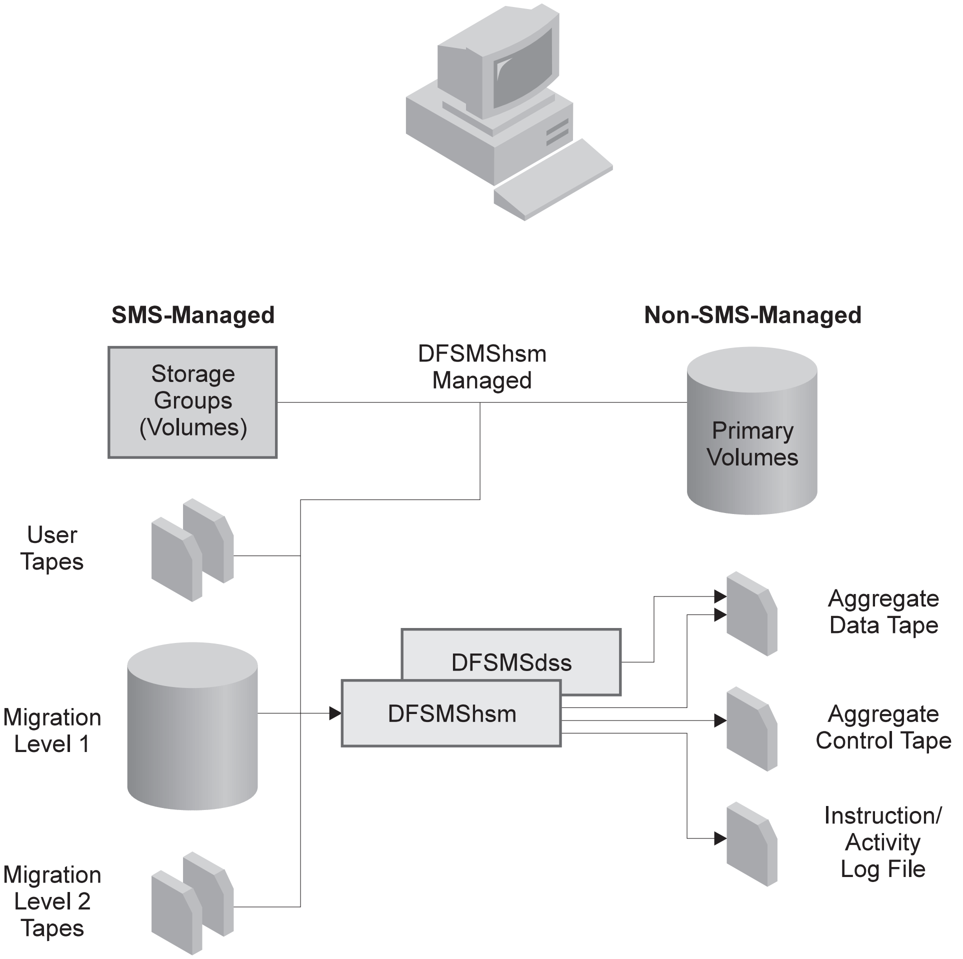


Figure 18. Aggregate Backup

Three types of aggregate backup tape files are created: an aggregate control file, an instruction/activity log file, and aggregate data files. The aggregate control file contains information needed at the remote site to recover the data sets. The instruction/activity log file contains either a copy of the instruction data set, a copy of the ABACKUP activity log, or both. The instruction data set contains instructions or other information to assist people in recovering and running the application. The aggregate data files contain the data sets you want to recover and can come from L0 DASD, migrated data, or user tape files.

Included in the control file information is the amount of storage required at each hierarchy level to recover the aggregate data sets. This information is automatically generated during backup processing.

You can adjust the performance of ABACKUP processing when dumping level 0 DASD by specifying whether you want one, two, or five tracks read at a time, or one cylinder at a time. You can also specify that ABACKUP output files from a single aggregate group are to be stacked on a minimum number of tape cartridges.

Recovering Aggregate Groups

During aggregate recovery, the backed-up data sets are recreated at the recovery site. Data sets backed up as migrated data sets can be returned to the same level as when backed up, or they can all be recovered to ML1 DASD, or they can all be recovered to ML2 tape. SMS-managed generation data sets being recovered to level 0 DASD can be recovered with a status of deferred, active, rolled off, or source, where source means restoring the data set to the status it had when backed up. All data sets backed up from user volumes are returned to user volumes.

A generation data group (GDG) base can be defined at the recovery site without having to back up any of the generation data sets (GDSs). The GDG base name is defined, if one doesn't already exist, during ARECOVER processing, and before any GDSs are recovered.

During aggregate recovery processing, the aggregate backup volumes can be automatically defined to RACF if the appropriate DFSMShsm commands have been specified and if RACF is active at the recovery site.

If data set name conflicts arise during aggregate recovery, a large variety of resolution options are available to handle the conflicts. Among the options are renaming all data sets with a new high-level qualifier, renaming only certain data sets with a new high-level qualifier, renaming data sets prior to recovery from the aggregate backup files, replacing the existing data set with the new data set, and bypassing the recovery of certain data sets. There is also a conflict resolution data set that can be edited by the user to handle conflicts that are not resolved by any of the other options. An edited conflict resolution data set can be used by subsequent aggregate recover processing to resolve conflicts.

To aid the user in the aggregate recovery process, two messages are produced: one to let the user know which data sets have been successfully recovered and the other to let the user know which data sets have failed recovery.

User Data Sets and Volumes

User data sets and volumes must be recovered for reasons other than disaster survival. DFSMShsm availability management uses the DFSMShsm incremental backup process, inline backup, and automatically invokes the DFSMSdss full physical volume dump process as a means of ensuring that the latest version can be recovered.

Automatic Invocation of DFSMSdss Full Volume Dump

DFSMShsm automatically invokes DFSMSdss functions to dump DFSMShsm-managed user volumes and ML1 volumes. Each DFSMShsm-managed and ML1 volume can be assigned a dump class that defines:

- When and how to dump the volume
- How often to dump the volume
- How long to keep the dump copy

DFSMShsm can dump up to five concurrent copies of a volume in five dump classes. At least two concurrent copies are required to provide an on-site and an off-site dump tape. Each group of up to five concurrent copies of a volume is known as a generation. DFSMShsm maintains up to 100 generations of each volume.

DFSMShsm can dump up to 15 volumes concurrently. Each dump task selects its own dump tapes. The storage administrator selects the maximum number of concurrent tasks allowed.

DFSMShsm creates and maintains an inventory of the dump copies created. Figure 19 shows the dump paths of data sets managed with and without SMS.

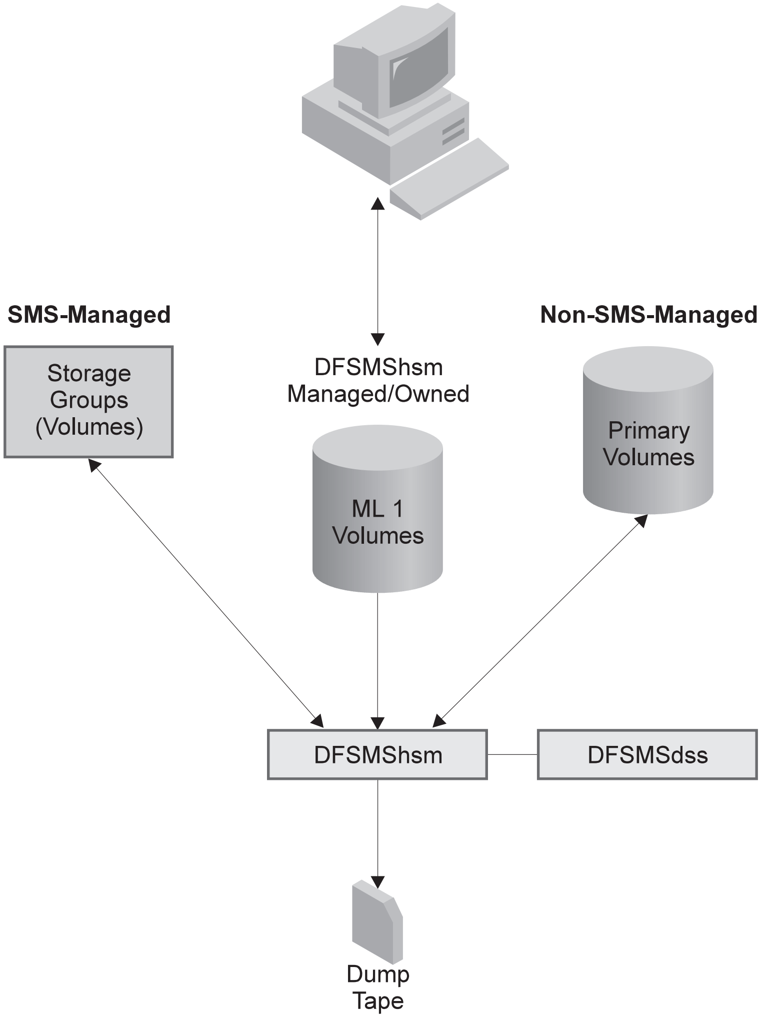


Figure 19. Automatic Dump and Restore

For system-managed volumes, DFSMShsm uses the automatic dump attributes and dump class names specified in the storage group definition.

For non-system-managed volumes, DFSMShsm uses the volume automatic dump attribute and dump class names specified for the volume.

Automatic DFSMShsm Incremental Backup

Backup of only new or changed user data sets is known as *incremental* backup. *Automatic incremental backup* ensures that up-to-date copies of new and changed data sets exist in case the original data sets are damaged or accidentally deleted. On the days and at the time specified by the storage administrator, DFSMShsm automatically copies new and changed data sets on DFSMShsm-managed volumes to DASD or tape backup volumes.

The DFSMShsm copy is a backup version that resides in an optionally compacted, device-independent format on a lower-cost-per-byte storage device. End-users can recover their own data sets without intervention by a storage administrator.

During the backup process, DFSMShsm prevents anyone from updating the data while the data set is being copied, unless the data set is backup-while-open eligible. If the data set is backup-while-open eligible, DFSMShsm invokes DFSMSdss to perform the incremental backup, but the data set change indicator is not reset, so the data set gets backed up again the next time backup is performed. DFSMShsm records the location of the backup version in its backup control data set (BCDS) and maintains records of its tape volumes in the offline control data set (OCDS).

DFSMShsm can back up a maximum of 15 user volumes concurrently. Each incremental backup task selects its own daily backup volume, either tape or DASD. The storage administrator selects the maximum number of concurrent tasks.

DFSMShsm allows automatic retention of up to 100 backup versions of each data set. DFSMShsm has the capability to determine and delete those backup versions that are no longer needed according to criteria supplied by the storage administrator.

Automatic incremental backup:

- Backs up user data sets on volumes that have been assigned the automatic backup attribute.
- Moves user backup versions created by command from temporary storage on DASD ML1 volumes to tape backup volumes.
- Backs up user data sets that migrated to DASD ML1 volumes with their change bits on. The change bit being on indicates that the data set has been changed since it was last backed up.
- Backs up the DFSMShsm control data sets and journals as a synchronized set.

Figure 20 on page 111 shows the backup paths of data sets managed with SMS and without SMS.

For system-managed volumes, DFSMShsm uses the storage group automatic backup attribute to determine what volumes to back up and the management class attributes to determine whether to back up a data set and how often to back it up.

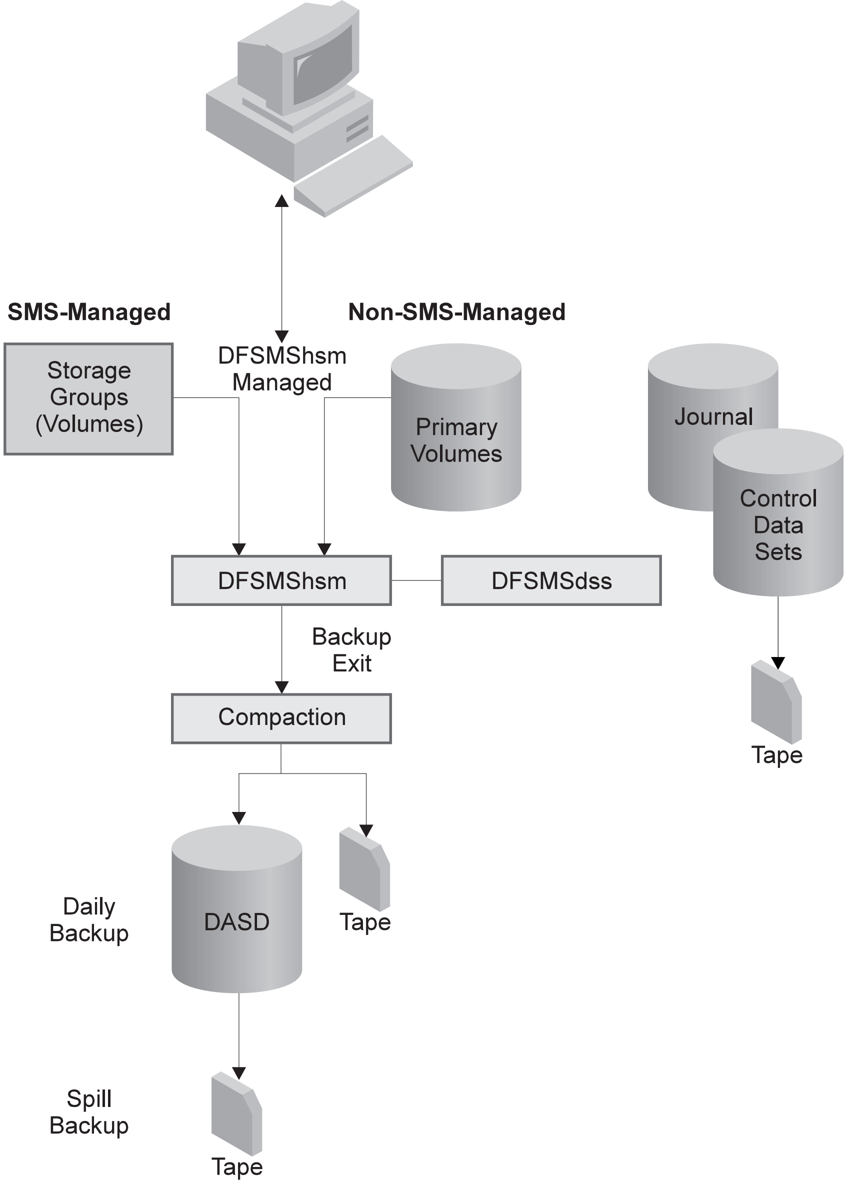


Figure 20. Automatic Incremental Backup

After the data set is backed up, DFSMShsm determines from the management class attributes how many versions to maintain and how long to keep them.

For non-system-managed volumes, DFSMShsm uses the volume attributes to determine what volumes to back up. If DFSMShsm's automatic backup attribute is active for the volume, all data sets requiring backup are backed up.

Inline Backup

You can back up a data set during job processing by invoking DFSMShsm to perform inline backup.

DFSMShsm provides two distinct methods to perform inline backup. Each has its benefits over the other. The first method is to used the HBACKDS command in a Terminal Monitor Program (TMP) batch job step. This method allows for data set filtering using % and * characters for masking, and provides an option to only perform the backup if the data set has been modified since the last backup was taken by DFSMShsm. The second method is to use the ACRINBACK program in a batch job step. Each data set is identified by a unique DD card and supports the use of JCL name substitution and relative DFD names. The second method can be used while the job has an exclusive enqueue on the data set, for example, if it was created by an earlier job step.

You request inline backup copies by placing the appropriate statements in a job step. The copies are created by DFSMShsm incremental backup during processing. Inline backup writes the backup version on a ML1 DASD volume to minimize unplanned tape mounts. The backup versions can be moved from ML1 to incremental backup volumes, automatically during incremental backup processing, or by command.

Recovering Data Sets

Data set recovery is the process of copying a backup version back to a user DASD volume. Anywhere from one to 64 individual data set recovery tasks can be performed concurrently, as defined by the installation.

A DFSMShsm-authorized storage administrator can enter a data set recover command to:

- Recover the latest data set version of any data set implicitly from a dump or an incremental backup volume
- Recover an explicit data set version from a dump or an incremental backup volume

End-users can enter data set recover commands to recover only their own data sets, either the latest or a specific version. The data recovered can come from either dump or incremental backup volumes.

Recovering DASD Volumes

Volume recovery is the process of copying all data back to a user DASD volume.

A DFSMShsm-authorized storage administrator can enter a volume recover command to:

- Restore a volume from the DFSMSdss dump copy with an option to recover individual data sets to the level of the most recent DFSMShsm incremental backup data set versions. This option scratches restored data sets that are now invalid because they:
  - Migrated or were scratched after the dump copy was made
  - Are uncataloged but have the same name as a cataloged data set
- Restore the volume only from the DFSMSdss dump copy
- Recover the volume only from the DFSMShsm incremental backup data set versions

The multitasking capabilities of individual data set recoveries can speed volume recoveries when incremental data set backup versions are being used in the recovery process. The incremental backup data sets are scheduled to be processed by data set recovery tasks, rather than being processed by the volume recover task. If more than one volume is being recovered, the incremental data set recovery requests for one volume can be scheduled, and the recovery process can immediately begin on the next volume, lessening the time needed for the total recovery process.

Maintaining the DFSMShsm Control Data Sets and Journal

To protect your inventory of backed up and migrated data, the three DFSMShsm control data sets and journal are backed up as a synchronized set at the start of automatic incremental backup. DFSMShsm maintains a user-specified number of backup and journal generations. The journal, a record of all transactions posted to the control data sets between backups, is used in the recovery process to update the recovered control data sets to the point-of-failure.

Although DFSMShsm provides the necessary facilities for recovering the control data sets, you can significantly improve the performance of recovery by using DFSORT to sort the journal before applying the journal changes to the control data sets. For more information on using DFSORT, see *DFSORT Application Programming Guide R14* .

Managing Availability Manually

In addition to automatic operations, DFSMShsm provides commands to perform availability operations on volumes and on data sets. The following tasks can be accomplished by command:

- Deletion of expired backup versions
- Duplication of DFSMShsm tapes (called alternates)
- Replacement of DFSMShsm tapes by alternates
- Volume dump
- Volume backup
- Data set backup

Some of the tasks can be started automatically or by command. Some can be started only by command.

Deletion of Expired Backup Versions

A command is available that determines expired backup versions of data sets and deletes them. Management class attributes are used to determine which expired backup versions of system-managed data sets are to be deleted. The management class attributes specify:

- The number of backup versions of existing data sets to keep
- The number of backup versions of deleted data sets to keep
- The number of days that versions other than the most recent, limited by the maximum to be kept, are allowed to be kept
- The number of days the last remaining backup version should be kept after DFSMShsm has determined that the user data set has been deleted

Deletion of expired backup versions of non-system-managed data sets is determined by parameters of the command. The command parameters can be used to delete:

- Backup versions of cataloged data sets after the specified number of days have elapsed since DFSMShsm has determined that the user data set had been deleted
- Backup versions of uncataloged data sets after the specified number of days have elapsed since the creation of the version
- Backup versions of retired data sets after the specified number of days have elapsed since the creation of that version

Duplication of DFSMShsm Tapes

Automatic and command dumps provide for offsite storage of data from level 0 and ML1 volumes, but not for backup and ML2 data that has been written to tapes. Off-site protection of data sets can be done with aggregate backup, volume dumping and TAPECOPY, or by using the duplex tape function.

DFSMShsm provides a command that allows the storage administrator to copy migration and backup tapes. The copies can provide for off-site storage or protection from media damage. To prepare backup copies of DFSMShsm backup and migration tapes for offsite storage, DFSMShsm-authorized users can enter a single command to cause DFSMShsm to make alternate copies of full, cartridge-type, single-file tapes. Options exist to copy all such DFSMShsm tapes, all ML2 tapes, or all backup tapes for which no alternate tape exists. Options also exist to copy a list of tapes, regardless of whether an alternate copy exists or whether the volumes are full.

DFSMShsm associates each alternate tape volume with the volume from which it was duplicated. If a volume that has an alternate volume is duplicated again, the old alternate volume is disassociated from the original volume and is replaced by the new alternate volume. Thus, DFSMShsm keeps records for only one alternate copy of each volume.

When an original volume becomes empty or is deleted from DFSMShsm control, any associated alternate volume is deleted from DFSMShsm’s records. You can use the DFSMSrmm component to manage the movement of alternate copies to an offsite location.

The DFSMShsm duplex tape function provides an alternative to the TAPECOPY function. The duplex tape function allows you to concurrently create two copies of backup and migration cartridge tapes. One copy becomes the original, the other the alternate tape. The alternate tape can be created onsite and moved to a remote location, or it can be written to a remote location. Both tapes must be created on tape drives and cartridges of the same type, such as 3490 tape drives with extended capacity cartridge storage tapes. The tapes are compatible with the tapes created by the TAPECOPY function.

Replacement of DFSMShsm Tapes with Alternates

If a DFSMShsm original backup or ML2 tape is damaged, the data from that tape can be retrieved from the alternate tape. A DFSMShsm-authorized user can enter a command that causes DFSMShsm to replace the references in its inventory to the original tape with the references to the alternate tape. Thus, when DFSMShsm refers to the data that was on the original tape, the data is retrieved from the tape that is the new original but was the alternate tape.

DFSMShsm also provides a function called disaster alternate. Disaster alternate allows a much faster form of tape replacement than previously existed. Should a true disaster occur, the user enters a command to cause DFSMShsm to flag all existing alternate tapes as disaster alternates and then another command to set DFSMShsm to run in disaster mode. When in disaster mode, each data set recall and recover causes an extra check of the volume record to see if a disaster alternate tape should be substituted for the original. Disaster alternate also allows an installation to test their disaster plan much more quickly than was previously possible.

Volume Dump

During volume dump, DFSMShsm uses a DFSMSdss function to back up the entire allocated space on a level 0 or DFSMShsm-owned volume. Volume dumps can only be performed by DFSMShsm-authorized users.

Volume Backup

To request a backup of some or all data sets on a volume managed by DFSMShsm, all primary volumes, or a user volume not managed by DFSMShsm, DFSMShsm-authorized users can enter a command to back up an entire volume. Keywords in the command let the user specify whether to back up every data set on the volume or whether to back up data sets selected from the criteria specified in the keywords. Although the command is available to back up every data set on the volume, the use of a command dump usually provides better performance for such an operation.

Only DFSMShsm-authorized users can enter a command to back up a level 0 volume.

Data Set Backup

To request a copy of a specific data set from a mounted volume, users can enter a BACKDS command to a specific data set. DFSMShsm backs up the data set but temporarily stores it on a ML1 volume. The data set does not have to reside on a volume managed by DFSMShsm.

For SMS-managed data sets, a management class backup attribute determines whether or not data sets in the management class are allowed to be backed up by command.

Chapter 12. DFSMShsm Facilities

To support the principal functions of space management and availability management, DFSMShsm provides and uses a number of facilities that make DFSMShsm more efficient and convenient to use.

This chapter describes what these facilities are and how they can be used.

Using Storage Groups and Volume Pools

DFSMShsm processes system-managed DASD volumes in SMS pool storage groups. A pool storage group is a set of volumes with free space threshold and paths to the systems within the complex. By placing volumes in a storage group, the requests of data sets that have common storage requirements can more easily be satisfied.

In a multiple-processing-unit environment, storage groups can be defined to allow DFSMShsm automatic functions to be performed for specific storage groups by specific processing units. This is in contrast to allowing storage groups to be processed by all processing units in a multiple-processing-unit environment.

DFSMShsm continues to support volume pools that are not managed by the Storage Management Subsystem. Volume pools depend on the volume from which the data set last migrated. If the volume from which the data set last migrated is a member of a pool, DFSMShsm recalls the data set to any currently online volume in the pool. If the volume from which the data set migrated is not a member of a pool, DFSMShsm uses its nonpooling algorithms for recalling the data set.

Volume pooling allows the inclusion of individual application or group volumes under DFSMShsm control. It provides the capability to manage independent pools of volumes according to specific, installation-defined requirements. Volume pooling also allows the gradual placement of more volumes into a larger pool, thus benefitting from the increased performance, improved DASD space utilization, and additional flexibility of the larger resource. This gives the installation the opportunity to reevaluate its current pooling structures for possible consolidation, while still taking advantage of the automated space management features of DFSMShsm.

Using Tape

DFSMShsm takes advantage of the single-file format and the compaction algorithms available on cartridge-type devices, such as the IBM 3480, 3490, 3490E, and 3590-1. The single-file format allows for better performance and utilization of cartridges since they can contain hundreds or thousands of data sets on a single tape cartridge. DFSMShsm uses the capacity of the cartridges and provides ways to handle contentions for different data sets residing on the same cartridge.

DFSMShsm creates backup and migration output for tapes in single-file format and supports only cartridge-type tape devices for the output. The CDS backup, the dump, and the ABARS functions support both reel-type and cartridge-type tapes for their functions. Data sets that have been backed up or migrated to reel-type tapes, or that are in multi-file format, can still be recovered or recalled from those tapes.

© Copyright IBM Corp. 1980, 1999

Reel-type tapes can also be recycled by DFSMShsm, with the new tapes being created as cartridge-type tapes.

DFSMShsm and Tape Utilization

DFSMShsm allows you to specify the desired level of tape utilization (fullness) for a backup or migration tape volume written in single-file format. You can use the full tape volume, meaning that DFSMShsm writes until reaching the physical end-of-volume (EOV). However, the recommended method is to specify a percentage to indicate how much of the tape should be used before considering it full. DFSMShsm then forces a logical end-of-volume with this method. The DFSMShsm default is to fill the tapes 97% full. By filling a tape only 97% full, you allow for slight variances in capacity between two similar tape cartridges, should you want to copy the entire cartridge to another cartridge. The second cartridge could then be used as an alternative source of the data should the first cartridge become damaged or unavailable.

Marking Partially-Filled Tapes as Full

When a backup or migration task ends, the last tape is usually only partially filled. When new processing begins, DFSMShsm selects that last partially-filled tape on which to continue processing. This lets you completely fill your tapes, but it does require a specific tape mount.

DFSMShsm allows you to specify that your partially filled tapes be marked as full. Then, when new processing begins, DFSMShsm issues a non-specific tape mount, which is beneficial when using cartridge loaders.

If you aren't using the duplex tape function, marking partially-filled tapes as full has another benefit. If you want to copy the migration and backup tapes for the purpose of disaster backup protection, marking a tape full allows it to be copied by the generic TAPECOPY command. A partially-filled tape that is not marked as full is not eligible to be copied. The data on the partially-filled tape would not have a disaster backup copy until the next time tape migration and backup processing occurred, and the tape was filled, and the TAPECOPY command was run.

If you *are* using the duplex tape function, even partially-filled migration and backup tapes have copies. Marking partially-filled tapes as full would still provide the benefit of non-specific tape mounts.

Each installation should weigh the benefits of marking partially-filled tapes as full against the benefits of utilizing the full storage capacity of the tapes, and decide what works best for them.

Automating DFSMShsm Tape Processing

Both cartridge loaders and automated tape libraries offer automated tape processing with DFSMShsm. The cartridge loaders are manually loaded and can hold up to multiple tape cartridges. An automated tape library is a library where tape cartridges are stored, selected, and loaded onto tape drives without manual intervention. It manages the tape volume itself, not the data sets on the tape volumes.

An automated tape library allows DFSMShsm to perform operations such as recycle, backup, dump, or migration to tape without any manual intervention. Recalls or recoveries of data sets or volumes that were migrated, backed up, or dumped to tape can also be done without manual intervention. When DFSMShsm issues a request for one or more tape volumes, the automated tape library locates the requested volumes, and automatically loads them on a tape drive. When processing is complete, the cartridges are returned to storage.

Manual Tape Library Support

DFSMShsm will continue to support environments with existing IBM 3495 Model M10 Manual Tape Library Dataserver Model 10.

Tape Mount Management

By implementing the tape mount management methodology, you can improve tape utilization and greatly reduce the number of tape mounts required for tape output data sets.

The volume mount analyzer reviews your tape mounts and creates reports that provide you with the information you need to effectively implement the recommended tape mount management strategy.

With tape mount management, you can have your ACS routines redirect data sets targeted to tape and have them written initially to a designated DASD storage group. Later the data sets can be migrated collectively to ML2 tape, enabling your installation to use the full storage capacity of your tape cartridges. At the same time, instead of having one tape mount for each tape output data set, you have one tape mount per quantity of data equal to the logical capacity of your tape cartridge. Not only is less time spent mounting tapes, or waiting for tape mounts, there is less tape drive contention, which can speed the processing of other jobs requiring tape resources.

For example, if your tape cartridge held an average of 100 output data sets, and you were using tape mount management, you would have 10 output tape mounts per 1000 output data sets. If you were not using tape mount management, you would have 1000 output tape mounts per 1000 output data sets. The reduction in tape mounts for your installation could be significant.

Functioning Independently of Device Type

DFSMShsm provides the capability to function independently of the device type to which data is being transferred. That is, when you define a volume to DFSMShsm, you specify its device type. DFSMShsm then does all the work of converting for different track capacities and different allocation specifications necessary to accommodate the data sets sent to various volumes. For most access methods, this device independence allows migrating or backing up a data set from one kind of device and recalling or recovering it to another kind of device.

Invoking DFSMShsm

You can invoke DFSMShsm processing through DFSMShsm commands, job control language (JCL), ISMF, and macros.

There are two categories of DFSMShsm commands, authorized commands and nonauthorized commands. For more detail, see “Protecting DFSMShsm Commands and Parameters” on page 121.

To make command processing of individual data sets as easy as possible, you can use ISMF's menu-driven panels to submit commands to DFSMShsm.

For more information about ISMF, see "Interactive Storage Management Facility and SMS-Managed Storage" on page 37. For more information on using DFSMShsm with ISMF, see the ISMF online help panels and *DFSMS/MVS Using ISMF*.

The following tasks can be invoked by your application programs using DFSMShsm:

- Migrating a data set
- Recalling a migrated data set
- Deleting a migrated data set
- Backing up a data set
- Extracting information about the existence of backup versions
- Recovering a backed up or dumped copy of a data set
- Deleting a backup of a data set
- Sending a command to DFSMShsm
- Freeing work space in storage

Controlling Access to DFSMShsm Tasks and Data

DFSMShsm protects data sets from unauthorized access by controlling access to data sets and tasks.

Protecting Data

Access can be controlled by the use of both passwords and security programs at the same time. However, system-managed data sets are not password protected. If a data set is password protected and security-program protected, DFSMShsm allows access to the data set without checking the password if the security program authorizes the access.

DFSMShsm also provides protection against unauthorized use or deletion of its owned tape volumes.

To provide security program protection, DFSMShsm calls the System Authorization Facility (SAF) when any unauthorized user enters a command that manipulates a data set or its backup copies.

As an installation option, users can submit batch jobs containing DFSMShsm commands in secure systems without RACF. DFSMShsm retrieves the user ID from the TSO protected step control block (PSCB) for a TSO batch request and associates it with the request so that authorization can be checked.

DFSMShsm optionally creates a backup profile for the most recent backup version of a cataloged data set if the data set is protected with a RACF discrete data set profile when it is backed up. DFSMShsm maintains only one backup profile for all the backup versions of the cataloged data set. The backup profile is used to re-create the discrete data set profile if it does not exist when the data set is recovered. When all backup versions of a data set are deleted, the related backup profile is also deleted.

DFSMShsm-owned data on DASD is named so that it can be protected by RACF generic profiles.

Protecting Data on Tapes

DFSMShsm provides three basic methods for protecting tape volumes under its control:

- RACF
- Expiration date
- Password

You can choose more than one tape protection option.

RACF

DFSMShsm protects each tape backup volume, dump tape volume, and tape migration level 2 (ML2) volume with RACF. The tape volume is RACF-protected when the tape is first written. DFSMShsm protects the tape volume with RACF by adding the volume to the RACF tape volume set for DFSMShsm. All tape volumes in the RACF tape volume set for DFSMShsm share the same access list and auditing controls. Protection of the tape volume is removed when the tape becomes empty and is returned to scratch status.

Expiration Date

If you use expiration date protection, DFSMShsm protects each tape backup volume, dump tape volume, tape ML2 volume, and aggregate tape volume with an expiration date. DFSMShsm places an expiration date of 99365 in the IBM Standard Data Set Label 1 unless specified otherwise by a DFSMShsm parameter or installation exit.

Password

If you use password protection, DFSMShsm causes each backup, dump, or migration tape to be password indicated. The password indication is placed in the security byte in the IBM Standard Data Set Label 1.

Protecting DFSMShsm Commands and Parameters

Because DFSMShsm operates as an MVS-authorized task, it can manage data sets automatically regardless of their password or security-program protection. To prevent unwanted changes of the parameters that control all data sets, DFSMShsm has classified its commands as authorized and nonauthorized. Authorized commands can be entered only by a user specifically authorized by the storage administrator. Generally, authorized commands can affect data sets not owned by the person issuing the command. Nonauthorized commands can be entered by any user, but they generally affect only those data sets for which the user has appropriate access authority.

Authorized users are assigned either USER or CONTROL authorization. Users with USER authorization can enter any DFSMShsm command except the one to authorize other users. Users with CONTROL authorization can enter any DFSMShsm command.

DASD Erase-on-Scratch Support

DFSMShsm provides the option to erase scratched data sets from its owned volumes under control of RACF. The erasure is under control of the original data set's RACF profile. See "Resource Access Control Facility Protection" on page 57 for additional information.

Preserving Data Set Integrity

While DFSMShsm is processing a data set, it protects that data set. Storage administrators can use management class attributes to prevent migration and backup of data sets. These attributes ensure data integrity by not allowing the data sets to be moved or copied with DFSMShsm migration or backup functions.

If data is moved or copied, DFSMShsm ensures that the migration or backup operation has been performed correctly before it erases any data set or resets its changed indicator. DFSMShsm ensures the integrity by using either volume reserves or global resource serialization capability. For more information about global resource serialization, see "Using SMS in a Multisystem Environment" on page 35.

Channel Path Load Balancing

During automatic volume space management, backup, and dump, DFSMShsm distributes the DFSMShsm-managed DASD volume processing work load to balance DFSMShsm's use of associated channel paths.

Chapter 13. Role of the Functional Component DFSMSrmm

DFSMSrmm is the functional component of DFSMS/MVS that helps you manage your removable media, such as tape cartridges, reels, and optical volumes. DFSMSrmm provides a central on-line inventory of the resources in your removable media library and in storage locations outside your removable media library.

A removable media library contains all the tape and optical volumes that are available for immediate use and includes the shelves where they reside. A removable media library usually includes other libraries: system-managed libraries and non-system-managed libraries, either automated or manual.

Storage locations are locations outside your removable media library. You use DFSMSrmm defined storage locations or define your own storage locations for managing disaster recovery or vital records or keeping track of volumes sent to other locations for processing.

DFSMSrmm manages your tape volumes within your removable media library at several levels:

Managing Shelves
Shelves are where you store your tape volumes. Using DFSMSrmm, you can more efficiently group your shelves and keep track of what volumes reside on them. For more information, see “Managing Shelves” on page 124.

Managing Volumes
DFSMSrmm helps you manage the movement and retention of your tape volumes over their full life. For more information, see “Managing Volumes” on page 124.

Managing Data Sets
DFSMSrmm records information about data sets on the volumes it manages and then uses that information to validate volumes. DFSMSrmm can also control the retention of data sets. For more information, see “Managing Data Sets” on page 125.

Refer to the following DFSMSrmm publications for more information:

- *DFSMS/MVS DFSMSrmm Application Programming Interface*
- *DFSMS/MVS DFSMSrmm Command Reference Summary*
- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*
- *DFSMS/MVS DFSMSrmm Diagnosis Guide*

Managing Storage with DFSMSrmm

DFSMSrmm helps you manage your tape volumes and shelves at your primary site and storage locations by recording information in a DFSMSrmm control data set.

You can manage storage at the shelf, volume, and data set level.

© Copyright IBM Corp. 1980, 1999

Managing Shelves

DFSMSrmm helps you manage the shelves in your tape library and storage locations, simplifying the tasks of your tape librarian. When you define a new volume in your library, DFSMSrmm shelf-manages the volume by automatically assigning it a place on the shelf, unless you request a specific place. DFSMSrmm does not perform[2] shelf-management for volumes that reside in virtual tape server. Also, your shelves are easier to use when DFSMSrmm manages them in pools. Pools allow you to divide your shelves into logical groups where you can store volumes. For example, you can have a different pool for each system your installation uses. Volumes to be used on each system are then stored together.

When you move volumes to a storage location, DFSMSrmm checks for available shelf space and then automatically assigns each volume a place on the shelf if you request it. You also have the option of defining your own storage locations and shelf space in those locations.

Managing Volumes

DFSMSrmm manages your volumes over their full life, from initial use to the time they are retired from service. When a tape volume is mounted, DFSMSrmm checks the volume label to ensure that the correct volume is mounted. DFSMSrmm then automatically records details about the volume, such as volume label and data set information. When tape volumes are released, DFSMSrmm can return them to scratch status, as well as performing other actions you request. For example, DFSMSrmm can automatically erase and initialize volumes upon release.

You do not have to define a tape volume to DFSMSrmm to use the volume. Volumes used to satisfy non-specific requests and all system-managed tape volumes must be managed by DFSMSrmm. Volumes in a system-managed tape library are automatically defined to DFSMSrmm upon entry into the system-managed tape library, if the volume is not previously defined. DFSMSrmm provides functions to control the use of undefined volumes and allows you to identify volumes that should be ignored.

To enhance the integrity of your volumes, DFSMSrmm identifies volumes with permanent and temporary errors. DFSMSrmm identifies tapes with permanent errors as requiring replacement. Tapes with temporary errors are tracked and reported. This information helps you identify volumes that should be replaced or removed from the library, and prevents their reuse as scratch volumes.

DFSMSrmm eases your operator's handling of volumes by updating drive displays and operator mount messages to indicate a volume's correct shelf location. DFSMSrmm supports the display of messages on drives that have displays, such as 3490s and 3590s. DFSMSrmm can update write-to-operator messages to show the volume's shelf location.

DFSMSrmm can also issue operator messages to a specific console name associated with a system-managed library. Messages are then issued to both the specific console and through standard message route codes.

[2] This support is available with APAR OW36342 or OW36343.

For control of volume movements and retention, you can define policies using DFSMSrmm vital record specifications for both volumes and data sets. For example, if you want to control the time a data set is retained before it is to be considered for release, you can define a vital record specification so that the volume expiration for the volume on which a data set resides is ignored. DFSMSrmm then uses the vital record specification retention criteria instead of the user specified retention period to retain the data set. Using DFSMSrmm vital record specifications, you have control over data set retention including whether or not user specified values are honored.

By using DFSMSrmm with the DFSMShsm functional component of DFSMS/MVS, you can extend tape management functions and tape integrity checking to DFSMShsm tape volumes. DFSMSrmm ensures that a private tape is not inadvertently mounted instead of a scratch tape. DFSMSrmm can also control the movement of DFSMShsm tape volumes selected as vital records, such as alternate volumes required for disaster recovery.

DFSMSrmm supports all the tape label types supported by DFSMSdfp and records the tape label type for each DFSMSrmm-managed volume. Support is included to allow DFSMSrmm to automatically track the use of non-standard label (NSL) tape volumes. DFSMSrmm also supports the use of bypass label processing (BLP). See “Magnetic Tape Volumes and Libraries” on page 74 for more information on tape labels. DFSMSrmm provides support for ISO/ANSI Version 4 tape labels.

DFSMSrmm does not support the management of duplicate volume serial numbers, but it does provide facilities that you can use to request DFSMSrmm to allow the use of these and other specific volumes.

Managing Data Sets

When a tape volume is used on the system, DFSMSrmm automatically records information, such as expiration dates, about each data set on the volume. DFSMSrmm uses these expiration dates to determine when it can release the volume.

DFSMSrmm provides options that you can use to retain data sets and volumes:

- By Cycles and By Days Cycle
- By Elapsed Days
- By Day Last Referenced
- While the data set is cataloged
- Until the expiration date is reached
- By Data Set Open or Abend Status
- By Job Name and Data Set Name
- By Generic Job Name and Generic Data Set Name Masks
- By Specific Volume Serial Number
- By Generic Volume Serial Number
- By Specific Date

DFSMSrmm prevents volumes containing unexpired data from being used as scratch. With DFSMSrmm, you can manage:

- Volumes that contain just one data set
- Volumes that contain multiple data sets
- Data sets that span more than one volume
- Data set collections that span multiple volumes

DFSMSrmm supports generic data set names as filter criteria for searching the control data set, making it easier to create lists of resources.

DFSMSrmm provides support for dates beyond the 20th century by ensuring that DFSMSrmm records all dates using a 4-digit year. DFSMSrmm also allows you to specify dates using the 4-digit year. DFSMSrmm provides the same support for dates as DFSMSdfp.

Using DFSMSrmm

Your tape librarian and storage administration group are the focal point for using DFSMSrmm. DFSMSrmm helps these and other users perform the tasks required for managing tape volumes. You can control which users can access the data defined to DFSMSrmm, as well as the level of access they have. DFSMSrmm also helps you create valuable reports and provides utilities to maintain your tape library.

Performing User Tasks

Although DFSMSrmm automates many of the tasks required to manage your removable media library, there are some tasks you can perform manually. These tasks include:

- Defining the resources in your removable media library, such as volumes, data sets, and shelves, to the DFSMSrmm control data set
- Updating this same information
- Registering software product volumes
- Obtaining information about these resources
- Requesting scratch volumes for private use
- Releasing volumes
- Requesting electronic notification when a volume expires
- Confirming volume movements and release actions

You can request these tasks using the DFSMSrmm ISPF dialog, the DFSMSrmm application programming interface, or the TSO command set that DFSMSrmm provides. Generally you use the DFSMSrmm ISPF dialog to perform DFSMSrmm tasks. You might want to use the TSO commands in procedures you write. You can use the dialog and command set in the foreground. You can also submit requests for batch processing.

Controlling Access to DFSMSrmm Tasks and Data

Working closely with RACF, DFSMSrmm can manage discrete RACF TAPEVOL profiles so that RACF can ensure that only authorized users can access data on volumes defined to DFSMSrmm.

DFSMSrmm uses the System Authorization Facility (SAF) to authorize the use of commands and functions to further secure your DFSMSrmm resources.

Creating Reports

DFSMSrmm provides report facilities and sample reports to help you create several types of reports. You can use the system management facilities (SMF) and identify the record types used for tracking when information in DFSMSrmm has been updated or when confidential volumes have been accessed. Using the SMF records as input, you can create reports describing access to designated volumes and data sets.

You can create inventory and movement reports for volumes managed by DFSMSrmm. The inventory reports help you audit the content of your library and storage locations. The movement reports identify volume movement through your library and storage locations. You can also request information from the DFSMSrmm control data set to use as input for creating your own reports using a tool such as DFSORT's ICETOOL utility. Using the DFSORT ICETOOL utility for DFSMSrmm reporting is easier with DFSORT R14 which includes a feature for using symbolic field names. You can use symbolic names for the DFSMSrmm SMF records, extract file records, and activity file records instead of using field offsets or data types to create reports.

Maintaining Your Removable Media Inventory

DFSMSrmm provides several utilities you can use to keep the information in your tape library current and to perform various tasks related to maintaining your tape media. Run these utilities using your existing scheduling facilities, such as OPC/ESA, so you can choose the best time and sequence to perform regularly required tasks.

DFSMSrmm utilities perform the following tasks:

- Maintain the integrity of the DFSMSrmm control data set. You can verify the records in the control data set, back up the data set, and restore it.
- Control and track volumes and data sets managed by vital record specifications.
- Manage the release of volumes that have reached their expiration date and are no longer retained by vital record specifications.
- Define and verify the volume movements between your library and storage locations.
- Erase and initialize volumes.

Appendix A. DFSMS/MVS Library and Related Publications

The tables included here describe information sources that can help you understand and use the functions of DFSMS/MVS. The tables are organized as follows:

DFSMS/MVS Library provides comprehensive information about DFSMS/MVS, including guidance in establishing storage management practices and implementing system-managed storage.

Storage Subsystem Library (SSL) provides comprehensive information about direct access storage devices (DASD) and storage control units (with and without cache).

Related Product Publications provide helpful information for tasks associated with DFSMS/MVS. Included are publications for the OS/390 operating system.

Information about electronic access and other ways you can acquire DFSMS/MVS books and related publications is also included here.

DFSMS/MVS Library

The DFSMS/MVS library contains information that helps you evaluate, install, customize, use, and diagnose the DFSMS/MVS product.

Because DFSMS/MVS includes the functions formerly provided in MVS/DFP Version 3, DFDSS Version 2, and DFHSM Version 2, the DFSMS/MVS library contains information from the libraries of these earlier products as well as information on newer functions exclusive to DFSMS/MVS.

DFSMS/MVS publications are available to you through the following channels:

- As a DFSMS/MVS license holder, you receive DFSMS/MVS publications as part of the distribution package containing the DFSMS/MVS program tapes and program directory. The IBM Distribution Center includes those publications required for you to plan, install, and customize functions of the current release of DFSMS/MVS (classified as basic materials).

  DFSMS/MVS Online Product Library is also shipped with your order. This CD-ROM includes unencrypted licensed materials and is available only to licensees of DFSMS/MVS.

- You can order additional copies of all DFSMS/MVS printed books or the CD-ROM from the IBM Distribution Center or your local branch office.
- DFSMS/MVS publications are available with these kits: *IBM Online Library Omnibus Edition MVS Collection* and *IBM Online Library Omnibus Edition OS/390 Collection*.
- If you subscribe to the S/390 Technical Library you can access selected DFSMS/MVS publications electronically through IBMLINK or on the Internet. See “Electronic Access to Online IBM Books” on page 139 for more information.

© Copyright IBM Corp. 1980, 1999

Publication	Order No.	Description of Contents	Audience
General Product Information			
DFSMS/MVS General Information	GC26-4900	Overview of DFSMS/MVS.	all users
DFSMS/MVS Library Guide	GC26-4902	Highlights of library changes for the current release; Cross-reference to DFSMS/MVS books and related product publications.	all users
Installation and Migration Planning			
DFSMS/MVS Licensed Program Specifications	GC26-4903	Warranty information for DFSMS/MVS.	varies
DFSMS/MVS Planning for Installation	SC26-4919	Guidance for planning the installation of DFSMS/MVS and migration to new functions of DFSMSdfp, DFSMShsm, DFSMSdss, and DFSMSrmm.	system programmer, system analyst, storage administrator
Customization			
DFSMS/MVS DFSMShsm Implementation and Customization Guide	SH21-1078	Guidance for creating and maintaining DFSMShsm-owned data sets, including migration and coexistence considerations.	system programmer, storage administrator
DFSMS/MVS DFSMSrmm Implementation and Customization Guide	SC26-4932	Information about implementing and customizing the functions of DFSMSrmm.	system programmer, storage administrator
DFSMS/MVS Installation Exits	SC26-4908	Instructions on exit routines and modules that extend or replace IBM-supplied functions.	system programmer, storage administrator
Storage Management			
MVS/ESA SML: Leading a Storage Administration Group	SC26-3126	Guidance for setting up a storage administration group and planning for system-managed storage.	data processing manager, storage administration group manager
DFSMS/MVS Implementing System-Managed Storage	SC26-3123	Tasks for migrating to system-managed storage and optimizing use of the storage management subsystem.	storage administrator, system programmer
MVS/ESA SML: Managing Data	SC26-3124	Techniques for establishing and enforcing data set policies, managing active and inactive data, and providing data set security.	storage administrator, system programmer
MVS/ESA SML: Managing Storage Groups	SC26-3125	Techniques for designing storage groups, making the transition to storage groups, and maintaining and monitoring the effectiveness of storage groups.	storage administrator, system programmer

DFSMS/MVS DFSMSdfp Storage Administration Reference	SC26-4920	Instructions for initializing and maintaining the storage management subsystem and for performing storage management tasks using ISMF panels.	storage administrator, system programmer
DFSMS/MVS DFSMShsm Storage Administration Guide	SH21-1076	DFSMShsm tasks for managing storage, including examples of storage administrator commands and results of DFSMShsm processing.	storage administrator, system programmer, operator
DFSMS/MVS DFSMShsm Storage Administration Reference	SH21-1075	DFSMShsm storage management commands, including syntax and coding techniques.	storage administrator, system programmer, operator
DFSMS/MVS DFSMShsm Storage Administration Reference Summary	SX26-3808	Summary of DFSMShsm storage management command syntax.	storage administrator, system programmer, operator
DFSMS/MVS Managing Data Availability	SC26-4928	Information about disaster prevention and the recovery of DFSMShsm data based on real experiences.	storage administrator, system programmer, application programmer
DFSMS/MVS DFSMSdss Storage Administration Guide	SC26-4930	DFSMSdss tasks for copying, dumping, and restoring data sets and DASD volumes.	storage administrator, system programmer, operator
DFSMS/MVS DFSMSdss Storage Administration Reference	SC26-4929	DFSMSdss commands, including syntax and coding examples.	storage administrator, system programmer, operator
DFSMS/MVS DFSMSrmm Command Reference Summary	SX26-6016	Summary of DFSMSrmm command syntax.	storage administrator, system programmer, tape librarian, operator, and DFSMSrmm end user
DFSMS/MVS DFSMSrmm Guide and Reference	SC26-4931	Instructions for using DFSMSrmm to manage removable media.	storage administrator, system programmer, tape librarian, operator
DFSMS/MVS DFSMSrmm Application Programming Interface	SC26-7272	storage administrator, system programmer, tape librarian, operator	
DFSMS/MVS Using the Volume Mount Analyzer	SC26-4925	Instructions for analyzing tape mounts with the volume mount analyzer.	storage administrator, system programmer
DFSMS Optimizer User's Guide and Reference	SC26-7047	Instructions for installing and using the DFSMS Optimizer for your storage system analysis.	performance analyst, storage administrator, capacity planner, financial analyst
DFSMS/MVS NaviQuest User's Guide	SC26-7194	Instructions for using the NaviQuest tool to migrate to SMS and maintain an SMS configuration.	storage administrator, system programmer

Catalog Management			
DFSMS/MVS Managing Catalogs	SC26-4914	Techniques for managing integrated catalog facility catalogs, VSAM catalogs, and OS CVOLs.	system programmer, storage administrator
DFSMS/MVS Access Method Services for ICF	SC26-4906	IDCAMS commands for using integrated catalog facility catalogs.	system programmer, storage administrator, operator
DFSMS/MVS Summary of Access Method Services for ICF	SX26-3807	Summary of IDCAMS commands for integrated catalog facility catalogs.	system programmer, storage administrator, operator
DFSMS/MVS Access Method Services for VSAM	SC26-4905	IDCAMS commands for using VSAM catalogs.	system programmer, operator

Data Management			
DFSMS/MVS Using ISMF	SC26-4911	Commands and techniques for using ISMF to analyze and manage data resources.	ISMF end user
DFSMS/MVS DFSMShsm Managing Your Own Data	SH21-1077	DFSMShsm user command syntax, conventions, and coding examples.	DFSMShsm end user
DFSMS/MVS DFSMShsm User Commands Reference Summary	SX26-3806	Summary of DFSMShsm user command syntax.	DFSMShsm end user
DFSMS/MVS Macro Instructions for Data Sets	SC26-4913	Access method macro instructions for processing user data sets.	application programmer, system programmer
DFSMS/MVS DFSMSdfp Advanced Services	SC26-4921	Techniques for extending DFSMS/MVS data management capabilities and other system data operations.	system programmer
DFSMS/MVS Using Data Sets	SC26-4922	Techniques for processing, protecting, and maintaining user data sets.	application programmer, system programmer
DFSMS/MVS Utilities	SC26-4926	Instructions for using DFSMSdfp utility programs for data, program, and device management.	application programmer, system programmer, storage administrator
DFSMS/MVS Checkpoint/Restart	SC26-4907	Information on establishing checkpoints during a program and restarting a job.	application programmer, system programmer
DFSMS/MVS Remote Copy Administrator's Guide and Reference	SC35-0169	Instructions for implementing remote copy as a disaster recovery and workload migration tool.	

Program Management			
DFSMS/MVS Program Management	SC26-4916	Instructions for using the binder, linkage editor, and loader to create, store, load, access, and update load modules and program objects.	application programmer, system programmer

Device Management			
DFSMS/MVS OAM Application Programmer's Reference	SC26-4917	Information about the OAM application interface used to manipulate objects.	application programmer
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support	SC26-4918	OAM planning, installation, customizing, and storage management information for object support using DASD, and optical and tape libraries.	storage administrator, system programmer
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries	SC26-3051	OAM planning, installation, customizing, and storage management information for tape libraries.	storage administrator, system programmer
DFSMS/MVS Using Magnetic Tapes	SC26-4923	Techniques for using magnetic tape volumes.	application programmer, storage administrator, system programmer

Distributed Data Access			
OS/390 Network File System User's Guide	SC26-7254	Instructions for accessing and creating MVS/ESA data sets from an AIX, UNIX, OS/2, or DOS environment using NFS protocol.	application programmer, system programmer
OS/390 Network File System Customization and Operation	SC26-7253	Techniques for customizing and operating the OS/390 Network File System server.	system programmer
DFSMS 1.3 Network File System Customization and Operations Guide	SC26-7029	Techniques for customizing and operating the DFSMS/MVS Network File System server.	system programmer
DFSMS/MVS DFM/MVS Guide and Reference	SC26-4915	Information about accessing and creating MVS/ESA data sets from remote systems.	application programmer, system programmer

Diagnosis (Restricted Materials)			
DFSMS/MVS DFSMSdfp Diagnosis Guide	SY27-9605	Information for diagnosing DFSMSdfp-related errors, including instructions for building keyword strings to search for known component failures.	system programmer, operator
DFSMS/MVS DFSMSdfp Diagnosis Reference	LY27-9606	Guidance for using DFSMSdfp diagnostic tools and aids to gather failure-related information.	system programmer, operator
DFSMS/MVS DFSMShsm Diagnosis Guide	LY27-9607	Diagnosis information for DFSMShsm, including instructions for building keyword search strings and other diagnostic tools.	system programmer, operator
DFSMS/MVS DFSMShsm Diagnosis Reference	LY27-9608	Information about the DFSMShsm control blocks and data areas used during diagnostic and maintenance procedures.	system programmer, operator
DFSMS/MVS DFSMSdss Diagnosis Guide	LY27-9609	Diagnosis information for DFSMSdss, including keyword search strings and other diagnostic tools.	system programmer, operator
DFSMS/MVS DFSMSrmm Diagnosis Guide	SY27-9615	Information for diagnosing DFSMSrmm-related errors, including how to use DFSMSrmm diagnostic tools.	system programmer, operator

Storage Subsystem Library

The Storage Subsystem Library (SSL) provides comprehensive information about direct access storage devices (DASD) and storage control units. In addition to describing characteristics, capabilities, and configurations for specific devices, the SSL specifies software requirements and options for using the devices in various operating environments. These publications complement books in the DFSMS/MVS library by providing device-specific techniques for effective storage management.

Publications from the SSL are shipped to you when you order these devices, and can be ordered separately as well. You can also receive SSL publications along with Device Support Facilities publications in BookManager readable format by ordering the *IBM Storage Subsystem Library and Device Support Facilities Release 14 Collection Kit* (SK2T-1209). The table here lists many of the books in the SSL. See your IBM representative for other device-related publications.

SSL Publication	Order No.	Description of Contents
Storage Subsystem Library Master Bibliography, Index, and Glossary	GC26-4496	Locating storage subsystem information.
IBM 3380 Direct Access Storage Introduction	GC26-4491	Overview of all IBM 3380 Models.
Using the IBM 3380 Direct Access Storage in an MVS Environment	GC26-4492	Planning, installing, and operating 3380s under MVS.
IBM 3390 Direct Access Storage Introduction	GC26-4573	Overview of all IBM 3390 Models.
Using IBM 3390 Direct Access Storage in an MVS Environment	GC26-4574	Planning, installing, and operating 3390s under MVS.
IBM 3990 Storage Control Introduction	GA32-0098	Introduction to the 3990.
IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide	GA32-0100	Planning, installing, operating, and using the IDCAMS cache commands with the 3990.
IBM 9340 Direct Access Storage Subsystems Introduction	GC26-4694	Introduction to the 9340.
Using IBM 9340 Direct Access Storage Subsystems in an MVS Environment	GC26-4646	Planning, installing, and operating 9340s under MVS.
IBM RAMAC Array DASD Introduction	GC26-7012	Introduction to RAMAC array DASD
Using IBM RAMAC Array DASD in MVS, VM, or VSE Environment	GC26-7013	Planning, installing, and operating RAMAC array DASD under MVS, VM, or VSE
IBM RAMAC Array Subsystem Introduction	GC26-7004	Introduction to the RAMAC array subsystem
Using IBM RAMAC Array Subsystem in MVS, VM, or VSE Environment	GC26-7005	Planning, installing, and operating the RAMAC array subsystem under MVS, VM, or VSE
IBM RAMAC Array Subsystem Reference	GC26-7006	Reference information on channel commands, sense bytes and error recovery procedures

Related Product Publications

The following publications are helpful for performing tasks related to, but not specifically part of, DFSMS/MVS. If a publication is not available through the IBM Distribution Center, contact your local IBM representative for assistance.

Publication	Order No.	Description of Contents
Online Library Collection Kits (CD-ROM)		
IBM Online Library Omnibus Edition MVS Collection	SK2T-0710	MVS-based publication libraries (including DFSMS/MVS library) in BookManager readable format.

System/390 Sysplex Publications		
OS/390 Parallel Sysplex Overview	GC28-1860	Introduction to the sysplex environment.
OS/390 Parallel Sysplex Hardware and Software Migration	GC28-1862	Planning for system-wide sysplex implementation.
OS/390 Parallel Sysplex Systems Management	GC28-1861	Planning for business, configuration, operations, and change management in a sysplex environment.
OS/390 Parallel Sysplex Application Migration	GC28-1863	Planning for migrating applications to an MVS/ESA sysplex environment.

MVS/ESA Publications		
OS/390 Introduction and Release Guide	GC28-1725	Overview of MVS/ESA SP and its requirements.
OS/390 Planning for Installation	GC28-1726	Requirements and planning information for an OS/390 installation.
OS/390 MVS Conversion Notebook	GC28-1747	Technical descriptions and migration actions for converting to the latest release of MVS/ESA.
OS/390 MVS Initialization and Tuning Guide	SC28-1751	Guidance for initializing MVS and improving system performance.
OS/390 MVS Initialization and Tuning Reference	SC28-1752	Reference to SYS1.PARMLIB members and selected system commands.
OS/390 JES2 Initialization and Tuning Reference	SC28-1792	Planning for installing, initializing, customizing and tuning JES2.
OS/390 JES3 Introduction	GC28-1808	Overview of JES3.
OS/390 JES3 Initialization and Tuning Reference	SC28-1803	Planning for installing, initializing, customizing and tuning JES3.
OS/390 HCD User's Guide	SC28-1848	Procedures for using HCD dialogs.
OS/390 MVS Planning: APPC/MVS Management	GC28-1807	Planning information for configuring, activating, customizing, and maintaining APPC/MVS.
OS/390 MVS Planning: Global Resource Serialization	GC28-1759	Planning information for using GRS to serialize MVS resources.
OS/390 MVS Setting Up a Sysplex	GC28-1779	Overview of setting up an MVS system to run in a sysplex.
OS/390 MVS System Management Facilities (SMF)	GC28-1783	Planning information for implementing SMF.
OS/390 MVS System Data Set Definition	GC28-1782	Instructions for defining system data sets, including those required for DFSMS/MVS.
OS/390 MVS JCL User's Guide	GC28-1758	Instructions for coding job control language.
OS/390 MVS JCL Reference	GC28-1757	Job control language reference
OS/390 MVS System Messages, Vol 1 (ABA-ASA)	GC28-1784	Message library for MVS/ESA products, including DFSMS/MVS.
OS/390 MVS System Messages, Vol 2 (ASB-EWX)	GC28-1785	
OS/390 MVS System Messages, Vol 3 (GDE-IEB)	GC28-1786	
OS/390 MVS System Messages, Vol 4 (IEC-IFD)	GC28-1787	
OS/390 MVS System Messages, Vol 5 (IGD-IZP)	GC28-1788	

Licensed Program Publications		
Assembler H Version 2 General Information	GC26-4035	Overview of Assembler H and its requirements.
IBM High Level Assembler/MVS & VM & VSE General Information	GC26-4943	Overview of High Level Assembler and its requirements.
Character Data Representation Architecture Overview	GC09-2207	Overview of CDRA and its applications.
Character Data Representation Architecture Reference and Registry	SC09-2190	Architecture and programming reference for CDRA.
CICS Transaction Server for OS/390 Migration Guide	GC34-5353	Migrating CICS applications.
CICS Recovery and Restart Guide	SC33–1698	Recovering and restarting CICS.
DB2 for MVS/ESA Version 4 Installation Guide	SC26-3456	Installing DB2 for MVS/ESA
Distributed Data Management Architecture: General Information	GC21-9527	Overview of DDM architecture.
DFSORT Getting Started R14	SC26-4109	Overview of DFSORT and its requirements.
DFSORT Application Programming Guide R14	SC33-4035	Instructions for using DFSORT to sort, merge, and copy data sets.
Device Support Facilities User's Guide and Reference	GC35-0033	Initializing DASD volumes, recovering from track errors.
Environmental Record Editing and Printing Program User's Guide and Reference	GC28-1378	Instructions for using EREP to record, edit, and print error records.
Integrated Cryptographic Service Facility/MVS General Information	GC23-0093	Overview of ICSF/MVS and its requirements.
OS/390 UNIX System Services Planning	SC28-1890	Planning information for setting up and managing OS/390 UNIX System Services.
OS/390 Security Server (RACF) Introduction	GC28-1912	Overview of RACF and its requirements.
OS/390 Security Server (RACF) General User's Guide	SC28-1917	Instructions for using RACF to perform basic security tasks.

Hardware Product Publications		
Advanced Function Printing: Printer Summary	G544-3135	Listing of AFP printers and their characteristics.
IBM 3494 Tape Library Dataserver Introduction and Planning Guide	GA32-0279	Planning for installation of the IBM 3494 Tape Library Dataserver including migration from non-automated tape to a fully automated tape library environment.
IBM 3495 Tape Library Dataserver Installation Planning and Migration Guide	GC35-0135	Planning for installation of the IBM 3495 Tape Library Dataserver including migration from non-automated tape to a fully automated tape library environment.
IBM 3995 Optical Library Dataserver Products: Introduction and Planning Guide	GA32-0121	Introduction to the IBM 3995 family of optical libraries and planning information for installing the machines.

ITSO Redbooks		
Get DFSMS FIT: Fast Implementation Techniques	SG24-2568	Information on using the DFSMS Fast Implementation Technique (FIT) for system-managed storage.
IBM Magstar Virtual Tape Server and Enhancements to Magstar: A New Era in Tape	SG24-4917	Description of the IBM Magstar Virtual Tape Server

Electronic Access to Online IBM Books

The S/390 Technical Library provides electronic access to over 5000 online IBM books in BookManager format.

The S/390 Technical Library is part of the System/390 Support Family of Services, a portfolio of services designed to complement IBM's software, hardware, and technical support services. Users can select S/390 Technical Library as one of the individual services to meet their specific requirements.

The S/390 Technical Library uses the BOOKS application on ServiceLink. The BOOKS application provides users with access to IBM product books available online. This access is provided electronically, through IBMLINK, from the ServiceLink main menu. The categories of available books are:

AIX	AIX/ESA
AS/400	AS/400
BULLETIN	US Technical Bulletins
CROSS PLATFORM	Cross Platform Applications (AD/Cycle, GDDM, DCF, etc.)
MVS/ESA	MVS/ESA (JES3, DFSMS, TSO/E, ISPF, SMP/E, DFSORT, etc.)
NETWORKING	Networking Systems (NCP, Netview, VTAM, TCP/IP, etc.)
PWS	Programmable Workstations (OS/2, LAN, MMPM/2, etc.)
REDBOOKS	ITSO Technical Bulletins
S/390 HARDWARE	Large Systems Hardware (ES/9000, ESCON, etc.)
TRANSACTION	Transaction Processing (CICS, IMS, DB2, QMF, OVVM, etc.)
VM, VSE	VM/ESA, VSE/ESA

Some of the functions available in BOOKS are the ability to select a category of books for reading, the ability to select a specific book to read, and the ability to do keyword searching in a single book or across a bookshelf of books. Form number and title searches are also allowed.

Internet Access

The S/390 Technical Library is now available on the Internet, off the IBM Home Page. Users who have subscribed to the S/390 Technical Library can access the service on the Internet by specifying the URL:

```
http://www1.ibmlink.ibm.com/cgi-bin/master
```

IBM Public Books

A selection of IBM technical books is available in online format on the Internet for access by all users. These books provide marketing support, planning and general information about products. You can access these books by specifying the URL given below and then selecting "Public Entry."

```
http://www1.ibmlink.ibm.com/cgi-bin/master
```

How To Subscribe

For more information about the S/390 Technical Library service and how to subscribe to it, contact your IBM representative or call the Support Family Information Center at 1-800-742-9235.

Abbreviations

The following abbreviations are defined as they are used in the DFSMS/MVS library.

This list may include acronyms and abbreviations from:

- *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.
- *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1).

ACDS. Active control data set.

ACS. Automatic class selection.

ADATA. Associated data.

ADSM. Adstar Distributed Storage Manager.

AIX. Advanced Interactive Executive.

AMODE. Addressing mode.

ANSI. American National Standards Institute.

AOR. Application owning region.

APF. Authorized program facility.

API. Application programming interface.

APPC. Advanced Program-to-Program Communication.

ATLDS. Automated Tape Library Dataserver.

BCDS. Backup control data set.

BCS. Basic catalog structure.

BDAM. Basic direct access method.

BLP. Bypass label processing.

BPAM. Basic partitioned access method.

BSAM. Basic sequential access method.

BTLS. Basic Tape Library Support.

CCSID. Coded Character Set Identifier.

CDS. Control data set.

CF. Coupling facility.

CFRM. Coupling facility resource manager.

CI. Control interval.

CICS. Customer Information Control System.

CICSVR. CICS VSAM Recovery.

COMMDS. Communications data set.

CUA. Common user access.

DADSM. Direct access device space management.

DASD. Direct access storage device.

DB2. Data Base 2.

DDM. Distributed Data Management.

DES. Data Encryption Standard.

DFM. Distributed FileManager.

DFSORT. Data Facility Sort.

DIV. Data in Virtual.

DSCB. Data set control block.

DSORG. Data set organization.

DTL. Data tag language.

EC. Extended control.

ELPA. Extended link pack area.

EOV. End-of-volume.

EPLPA. Extended pageable link pack area.

ESA. Enterprise Systems Architecture

ESCON. Enterprise System Connection.

ESD. External symbol dictionary.

ESDS. Entry-sequenced data set.

EXCP. Execute channel program.

© Copyright IBM Corp. 1980, 1999

FCB. Forms control buffer.

FIPS. Federal Information Processing Standard.

FLPA. Fixed link pack area.

FOR. File owning region.

GB. Gigabyte.

GDG. Generation data group.

GDS. Generation data stream.

GSAM. Generalized sequential access method.

GSR. Global shared resources.

GUI. Graphical user interface.

HCD. Hardware configuration definition.

HIDAM. Hierarchic indexed direct access method.

ICF. Integrated catalog facility.

ICKDSF. Device Support Facilities.

IDR. Identification record.

IMS. Information Management System.

I/O. Input/output.

IPL. Initial program load.

ISAM. Indexed sequential access method.

ISMF. Interactive Storage Management Facility.

ISO. International Organization for Standardization.

ISPF. Interactive System Productivity Facility.

JCL. Job control language.

JES. Job entry subsystem.

KB. Kilobyte.

KSDS. Key-sequenced data set.

LDMI. Local Data Management Interface.

LDS. Linear data set.

LPA. Link pack area.

LSR. Local shared resources.

MB. Megabyte.

MDS. Main device scheduling.

ML1. Migration level 1.

ML2. Migration level 2.

MLPA. Modified link pack area.

MVS. Multiple Virtual Storage.

NCSC. U.S. Department of Defense National Computer Security Center.

NFS. OS/390 Network File System.

NIST. U.S. National Institute of Standards and Technology.

NL. Non-labeled.

NSL. Non-standard label.

NSR. Non-shared resources.

OAM. Object Access Method.

OCDS. Offline control data set.

OLTP. Online transaction processing.

OSAM. Overflow sequential access method.

PDS. Partitioned data set.

PDSE. Partitioned data set extended.

PLPA. Pageable link pack area.

PPRC. Peer-to-peer remote copy.

PSCB. Protected step control block.

PSF. PSF for OS/390

PSP. Program Services Period.

PTF. Program Temporary Fix.

QSAM. Queued sequential access method.

RACF. Resource Access Control Facility.

RBA. Relative byte address.

REA. RAMAC Electronic Array Storage.

RETAIN. REmote Technical Assistance and Information Network.

RLD. Relocation dictionary.

RLS. Record-level sharing.

RMF. Resource Measurement Facility.

RMODE. Residence mode.

RRDS. Relative-record data set.

RSA. RAMAC Scalable Array Storage.

RSECT. Read-only control section.

RVA. RAMAC Virtual Array Storage.

SAA. Systems Application Architecture.

SAF. System Authorization Facility.

SCDS. Source control data set.

SDSP. Small data set packing.

SHCDS. Sharing control data set.

SMF. System management facility.

SMS. Storage Management Subsystem or System Managed Storage.

SNA. Systems Network Architecture.

SPUFI. SQL Processing Using File Input.

SQL. Structured query language.

SSL. Storage Subsystem Library.

TB. Terabyte.

TCP/IP. Transmission Control Protocol/Internet Protocol.

TMM. Tape mount management.

TMP. Terminal Monitor Program.

TP. Transaction program.

TSO. Time Sharing Option.

UCB. Unit control block.

UIM. Unit information module.

VIO. Virtual I/O.

VRS. Vital record specification.

VRRDS. Variable-length relative-record data set.

VSAM. Virtual Storage Access Method.

VTOC. Volume table of contents.

VTS. Virtual tape server.

VVDS. VSAM volume data set.

WTO. Write-to-operator.

XRC. Extended remote copy.

Glossary

The following terms are defined as they are used in the DFSMS/MVS Library. If you do not find the term you are looking for, see the IBM Software Glossary:

`http://www.networking.ibm.com/nsg/nsgmain.htm`

This glossary is an ever-evolving document that defines technical terms used in the documentation for many IBM software products.

A

access method services. A multifunction service program that manages VSAM and non-VSAM data sets, as well as integrated catalog facility (ICF) and VSAM catalogs. Access method services provides the following functions:

- defines and allocates space for VSAM data sets, VSAM catalogs, and ICF catalogs
- converts indexed-sequential data sets to key-sequenced data sets
- modifies data set attributes in the catalog
- reorganizes data sets
- facilitates data portability among operating systems
- creates backup copies of data sets
- assists in making inaccessible data sets accessible
- lists the records of data sets and catalogs
- defines and builds alternate indexes
- converts CVOLS and VSAM catalogs to ICF catalogs

activate. To load the contents of a source control data set (SCDS) into Storage Management Subsystem address space storage and into an active control data set (ACDS), or to load the contents of an existing ACDS into subsystem address space storage. This establishes a new storage management policy for the subsystem complex.

active configuration. The most recently activated SCDS, which now controls storage management for the Storage Management Subsystem complex.

active control data set (ACDS). A VSAM linear data set that contains an SCDS that has been activated to control the storage management policy for the installation. When activating an SCDS, you determine which ACDS will hold the active configuration (if you have defined more than one ACDS). The ACDS is shared by each system that is using the same SMS configuration to manage storage. See also *source control data set* and *communications data set*.

active data. (1) Data that can be accessed without any special action by the user, such as data on primary storage or migrated data. Active data also can be stored on tape volumes. (2) For tape mount management, application data that is frequently referenced, small in size, and managed better on DASD than on tape. Contrast with *inactive data*.

actual UCB. The UCB used for all I/O operations. It has an address that is consistent in any address space. The actual UCB can reside in common storage either above or below 16 MB.

aggregate backup. The process of copying an aggregate group and recovery instructions so that a collection of data sets can be recovered later as a group.

aggregate group. A collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy.

alternate index. In VSAM, a collection of index entries related to a given base cluster and organized by an alternate key, that is, a key other than the prime key of the associated base cluster data records; it gives an alternate directory for finding records in the data component of a base cluster.

always call. See *RACF always call*.

AMASPZAP. A service program used to used to dynamically update or dump programs and data sets.

AMBLIST. A service program used to print formatted listings of modules and system storage areas to aid in problem diagnosis.

American National Standards Institute (ANSI). An organization that establishes voluntary industry standards for information processing, particularly for control characters and magnetic tape labels.

application programming interface (API). A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

automated tape library. A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. See also *tape library*. Contrast with *manual tape library*.

© Copyright IBM Corp. 1980, 1999

automatic backup. (1) In DFSMShsm, the process of automatically copying data sets from primary storage volumes or migration volumes to backup volumes. (2) In OAM, the process of automatically copying objects from DASD, optical, or tape volumes contained in an object storage group, to backup volumes contained in an object backup storage group.

automatic class selection (ACS) routine. A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

automatic dump. In DFSMShsm, the process of using DFSMSdss automatically to do a full-volume dump of all allocated space on a primary storage volume to designated tape dump volumes.

automatic primary space management insert. In DFSMShsm, the process of deleting expired data sets, deleting temporary data sets, releasing unused space, and migrating data sets from primary storage volumes automatically.

automatic secondary space management. In DFSMShsm, the process of automatically deleting expired migrated data sets, deleting expired records from the migration control data sets, and migrating eligible data sets from migration level 1 volumes to migration level 2 volumes.

automatic volume space management. In DFSMShsm, the process that includes automatic primary space management and interval migration.

availability. For a storage subsystem, the degree to which a data set or object can be accessed when requested by a user.

B

backup. The process of creating a copy of a data set or object to be used in case of accidental loss.

backup control data set (BCDS). In DFSMShsm, a VSAM key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMShsm.

backup-while-open (BWO). This makes a backup copy of a data set while the data set is open for update. The backup copy can contain partial updates.

base configuration. The part of an SMS configuration that contains general storage management attributes, such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration manages.

binder. The DFSMS/MVS program that processes the output of language translators and compilers into an executable program (load module or program object). It replaces the linkage editor and batch loader in MVS

block count. The number of data blocks on a magnetic tape volume.

buffer. A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another.

C

cache fast write. A storage control capability in which the data is written directly to cache without using nonvolatile storage. Cache fast write is useful for temporary data or data that is readily recreated, such as the sort work files created by DFSORT. Contrast with *DASD fast write*.

cache set. A parameter on storage class and defined in the base configuration information that maps a logical name to a set of CF cache structure names.

capacity planning. The process of forecasting and calculating the appropriate amount of physical computing resources required to accommodate an expected workload.

captured UCB. A virtual window into the actual UCB which resides in private storage below 16 MB. All the virtual windows on the actual UCB see the same data at the same time. Only actual UCBs above the 16 MB line are captured.

Cartridge System Tape. The base tape cartridge media used with 3480 or 3490 Magnetic Tape Subsystems. Contrast with *Enhanced Capacity Cartridge System Tape*.

Catalog Search Interface. An application programming interface (API) to the catalog accessible from assembler and high-level languages. As an alternative to LISTCAT, it allows tailoring of output, provides additional information not provided by LISTCAT, while requiring less I/O than LISTCAT, because of using generic locates.

Character Data Representation Architecture (CDRA) API. A set of identifiers, services, supporting resources, and conventions for consistent representation, processing, and interchange of character data.

class transition. An event that brings about change to an object's service-level criteria, causing OAM to invoke

ACS routines to assign a new storage class or management class to the object.

client. (1) A function that requests services from a server, and makes them available to the user. (2) An address space in MVS that is using TCP/IP services. (3) A term used in an environment to identify a machine that uses the resources of the network. See also *source*.

client-server relationship. Any process that provides resources to other processes on a network is a *server*. Any process that employs these resources is a *client*. A machine can run client and server processes at the same time.

cluster. In VSAM, a named structure consisting of a group of related components. For example, when the data is key-sequenced, the cluster contains both the data and the index components.

Coded Character Set Identifier (CCSID). A 16-bit number that identifies a specific encoding scheme identifier, character set identifiers, code page identifiers, and additional coding required information. The CCSID uniquely identifies the coded graphic character representation used.

communications data set (COMMDS). The primary means of communication among systems governed by a single SMS configuration. The COMMDS is a VSAM linear data set that contains the name of the ACDS and current utilization statistics for each system-managed volume, which helps balance space among systems running SMS. See also *active control data set* and *source control data set*.

compatibility mode. For DFSMS/MVS, it is the mode of running SMS in which no more than eight names—representing systems, system groups, or both—are supported in the SMS configuration. When running in this mode, the DFSMS/MVS system can share SCDSs, ACDSs and COMMDSs with other systems running MVS/DFP or DFSMS/MVS releases prior to DFSMS/MVS 1.3, and with other DFSMS/MVS systems running in compatibility mode.

compress. (1) To reduce the amount of storage required for a given data set by having the system replace identical words or phrases with a shorter token associated with the word or phrase. (2) To reclaim the unused and unavailable space in a partitioned data set that results from deleting or modifying members by moving all unused space to the end of the data set.

concurrent copy. A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with the usual application program processing.

configuration (Storage Management Subsystem). A base configuration, definitions of Storage Management Subsystem classes and storage groups, and automatic class selection routines that DFSMS uses to manage storage.

connectivity. (1) The considerations regarding how storage controls are joined to DASD and processors to achieve adequate data paths (and alternative data paths) to meet data availability needs. (2) In a DFSMS environment, the system status of volumes and storage groups.

construct. One of the following: data class, storage class, management class, storage group, aggregate group, base configuration.

control data set (CDS). With respect to the Storage Management Subsystem, a VSAM linear data set containing configurational, operational, or communication information. The Storage Management Subsystem introduces three types of control data sets that guide the execution of the Storage Management Subsystem: the source control data set, the active control data set, and the communications data set.

control interval (CI). A fixed-length area of auxiliary storage space in which VSAM stores records. It is the unit of information (an integer multiple of block size) transmitted to or from auxiliary storage by VSAM.

coupling facility (CF). The hardware that provides high-speed caching, list processing, and locking functions in a Parallel Sysplex.

coupling facility (CF) cache structure. The CF hardware that provides a data cache.

coupling facility (CF) lock structure. The CF hardware that supports sysplex-wide locking.

D

DASD fast write. An extended function of some models of the IBM 3990 Storage Control in which data is written concurrently to cache and nonvolatile storage and automatically scheduled for destaging to DASD. Both copies are retained in the storage control until the data is completely written to the DASD, providing data integrity equivalent to writing directly to the DASD. Use of DASD fast write for system-managed data sets is controlled by storage class attributes to improve performance. See also *dynamic cache management*. Contrast with *cache fast write*.

DASD volume. A DASD space identified by a common label and accessed by a set of related addresses. See also *volume*, *primary storage*, *migration level 1*, *migration level 2*.

data class. A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

Data Facility Sort (DFSORT). An IBM licensed program that is a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, and copying operations, as well as providing versatile data manipulation at the record, field, and bit level.

data set. In DFSMS/MVS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In OS/390 non-UNIX environments, the terms *data set* and *file* are generally equivalent and sometimes are used interchangeably. See also *file*. In OS/390 UNIX environments, the terms *data set* and *file* have quite distinct meanings.

data set collection. A group of data sets which are intended to be allocated on the same tape volume or set of tape volumes as a result of data set stacking.

data set stacking. The function used to place several data sets on the same tape volume or set of tape volumes. It increases the efficiency of tape media usage and reduces the overall number of tape volumes needed by allocation. It also allows an installation to group related data sets together on a minimum number of tape volumes, which is useful when sending data offsite.

default device geometry. Part of the SMS base configuration, it identifies the number of bytes per track and the number of tracks per cylinder for converting space requests made in tracks or cylinders into bytes, when no unit name has been specified.

default management class. Part of the SMS base configuration, it identifies the management class that should be used for system-managed data sets that do not have a management class assigned.

default unit. Part of the SMS base configuration, it identifies an esoteric (such as SYSDA) or generic (such as 3390) device name. If a user omits the UNIT parameter on the JCL or the dynamic allocation equivalent, SMS applies the default unit if the data set has a disposition of MOD or NEW and is *not* system-managed.

device category. A storage device classification used by SMS. The device categories are as follows SMS-managed DASD, SMS-managed tape, non-SMS-managed DASD non-SMS-managed tape.

device management. The task of defining input and output devices to the operating system, and then controlling the operation of these devices.

Device Support Facilities (ICKDSF). A program used for initialization of DASD volumes and track recovery.

DFSMS environment. An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. In the DFSMS environment for MVS, the function is provided by DFSORT, RACF, and the combination of DFSMS/MVS and MVS.

DFSMSdfp. A DFSMS/MVS functional component or base element of OS/390, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS/MVS functional component or base element of OS/390, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS/MVS functional component or base element of OS/390, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMShsm control data set. In DFSMShsm, one of three VSAM key-sequenced data sets that contain records used in DFSMShsm processing. See also *backup control data set*, *migration control data set*, and *offline control data set*.

DFSMShsm-managed volume. (1) A primary storage volume, which is defined to DFSMShsm but which does not belong to a storage group. (2) A volume in a storage group, which is using DFSMShsm automatic dump, migration, or backup services. Contrast with *system-managed volume* and *DFSMSrmm-managed volume*.

DFSMShsm-owned volume. A storage volume on which DFSMShsm stores backup versions, dump copies, or migrated data sets.

DFSMS/MVS. An IBM System/390 licensed program that provides storage, data, and device management functions. When combined with MVS/ESA SP Version 5 it composes the base MVS/ESA operating environment. DFSMS/MVS consists of DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm.

DFSMS/MVS Network File System. See *OS/390 Network File System*.

DFSMS/MVS Optimizer Feature. A DFSMS/MVS feature that provides an analysis and reporting capability for SMS and non-SMS environments.

DFSMSrmm. A DFSMS/MVS functional component or base element of OS/390, that manages removable media.

DFSMSrmm-managed volume. A tape volume that is defined to DFSMSrmm. Contrast with *system-managed volume* and *DFSMShsm-managed volume*.

dictionary. A table that associates words, phrases, or data patterns to shorter tokens. The tokens replace the associated words, phrases, or data patterns when a data set is compressed.

direct access device space management (DADSM). A collection of subroutines that manages space on disk volumes. The subroutines are: Create, Scratch, Extend, and Partial Release.

disaster recovery. A procedure for copying and storing an installation's essential business data in a secure location, and for recovering that data in the event of a catastrophic problem. Compare with *vital records*.

Distributed Data Management (DDM). A data protocol architecture for data management services across distributed systems in an SNA environment. DDM provides a common data management language for data interchange among different IBM system platforms.

Distributed FileManager/MVS. (1) The term used to describe the SAA architectures and programming support that provide distributed file access capabilities between SAA systems. (2) The DFSMS/MVS component that implements the DDM target server.

dual copy. A high availability function made possible by nonvolatile storage in some models of the IBM 3990 Storage Control. Dual copy maintains two functionally identical copies of designated DASD volumes in the logical 3990 subsystem, and automatically updates both copies every time a write operation is issued to the dual copy logical volume.

dump class. A set of characteristics that describes how volume dumps are managed by DFSMShsm.

duplexing. The process of writing two sets of identical records in order to create a second copy of data.

dynamic cache management. A function that automatically determines which data sets will be cached based on the 3990 subsystem load, the characteristics of the data set, and the performance requirements defined by the storage administrator.

E

Enhanced Capacity Cartridge System Tape. Cartridge system tape with increased capacity that can only be used with 3490E Magnetic Tape Subsystems. Contrast with *Cartridge System Tape*.

erase-on-scratch. The physical erasure of data on a DASD data set when the data set is deleted (scratched).

expiration. The process by which data sets or objects are identified for deletion because their expiration date or retention period has passed. On DASD, data sets and objects are deleted. On tape, when all data sets have reached their expiration date, the tape volume is available for reuse.

extended addressability. The ability to create and access a VSAM data set that is greater than 4 GB in size. Extended addressability data sets must be allocated with DSNTYPE=EXT and EXTENDED ADDRESSABILITY=Y.

extended format. The format of a data set that has a data set name type (DSNTYPE) of EXTENDED. The data set is structured logically the same as a data set that is not in extended format but the physical format is different. See also *striped data set* and *compressed format*.

extended link pack area (ELPA). The extension of the link pack area that resides above 16 MB in virtual storage. See also *link pack area*.

extended pageable link pack area (EPLPA). The extension of the pageable link pack area that resides above 16 MB in virtual storage. See also *pageable link pack area*.

extended remote copy. Extended Remote Copy (XRC) is a technique involving both the DFSMS/MVS host and the I/O Subsystem that keeps a "real time" copy of designated data at another location. Updates to the primary center are replicated at the secondary center asynchronously.

F

file. A collection of information treated as a unit. In non-OS/390 UNIX environments, the terms *data set* and *file* are generally equivalent and are sometimes used interchangeably. See also *data set*.

file system. In the OS/390 UNIX HFS environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See also *HFS data set*.

filtering. The process of selecting data sets based on specified criteria. These criteria consist of fully or partially-qualified data set names or of certain data set characteristics.

G

giga (G). The information-industry meaning depends upon the context:

1. G = 1,073,741,824(2^{30}) for real and virtual storage
2. G = 1,000,000,000 for disk storage capacity (e.g. 4 Gb fixed disk)
3. G = 1,000,000,000 for transmission rates

global resource serialization (GRS). A component of MVS used for serializing use of system resources and for converting hardware reserves on DASD volumes to data set enqueues.

GRS complex (GRSplex). One or more MVS images that share a common global resource serialization policy in either a ring or star configuration.

group. (1) With respect to partitioned data sets, a member and the member's aliases that exist in a PDS or PDSE, or in an unloaded PDSE. (2) A collection of users who can share access authorities for protected resources.

H

hardware configuration definition (HCD). An interactive interface in MVS that enables an installation to define hardware configurations from a single point of control.

hierarchical file system (HFS) data set. A data set that contains a POSIX-compliant file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using OS/390 UNIX System Services. See also*file system.*

Hiperspace. A high performance space backed by either expanded storage or auxiliary storage, which provides high performance storage and retrieval of data.

HSM complex (HSMplex). One or more MVS images running DFSMShsm that share a common set of control data sets (MCDS, BCDS, OCDS, and Journal).

I

improved data recording capability (IDRC). A recording mode that can increase the effective cartridge data capacity and the effective data rate when enabled and used. IDRC is always enabled on the 3490E Magnetic Tape Subsystem.

inactive configuration. A configuration contained in an SCDS. A configuration that is not currently being used by the Storage Management Subsystem.

inactive data. (1) A copy of active data, such as vital records or a backup copy of a data set. Inactive data is never changed, but can be deleted or superseded by another copy. (2) In tape mount management, data that is written once and never used again. The majority of this data is point-in-time backups. (3) Objects infrequently accessed by users and eligible to be moved to the optical library or shelf. Contrast with *active data*.

indexed VTOC. A volume table of contents with an index that contains a list of data set names and free space information, which allows data sets to be located more efficiently.

in-place conversion. The process of bringing a volume and the data sets it contains under the control of SMS without data movement, using DFSMSdss.

integrated catalog facility catalog. A catalog that is composed of a basic catalog structure (BCS) and its related volume tables of contents (VTOCs) and VSAM volume data sets (VVDSs). See also *basic catalog structure* and *VSAM volume data set*.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS/MVS that allows users and storage administrators access to the storage management functions.

interval migration. In DFSMShsm, automatic migration that occurs when a threshold level of occupancy is reached or exceeded on a DFSMShsm-managed volume, during a specified time interval. Data sets are moved from the volume, largest eligible data set first, until the low threshold of occupancy is reached.

ISO/ANSI. When referring to magnetic tape labels and file structure, any tape that conforms to certain standards established by the ISO and ANSI. Tapes are sometimes called:

- **Version 3 tapes** Conforms to ISO 1001–1979 level 4, and ANSI X3.27–1978 level 4
- **Version 4 tapes** Conforms to ISO 1001–1986(E) LEVEL 4, ANSI X3.27–1987 level 4

J

JES3. An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job enqueue.

K

kilo (K). The information-industry meaning depends upon the context:

1. K = 1024(2^{10}) for real and virtual storage
2. K = 1000 for disk storage capacity (e.g. 4000 Kb fixed disk)
3. K = 1000 for transmission rates

key-sequenced data set (KSDS). A VSAM data set whose records are loaded in ascending key sequence and controlled by an index.

L

linear data set (LDS). A VSAM data set that contains data but contains no control information. A linear data set can be accessed as a byte-addressable string in virtual storage.

link pack area (LPA). In MVS, an area of virtual storage that contains reenterable routines that are loaded at IPL time and can be used concurrently by all tasks in the system.

load module. An executable program stored in a partitioned data set program library. See also *program object*.

logical storage. With respect to data, the attributes that describe the data and its usage, as opposed to the physical location of the data.

M

mega (M). The information-industry meaning depends upon the context:

1. M = 1,048,576(2^{20}) for real and virtual storage
2. M = 1,000,000 for disk storage capacity (e.g. 4000 Mb fixed disk)
3. M = 1,000,000 for transmission rates

management class. A collection of management attributes, defined by the storage administrator, used to control the release of allocated but unused space; to control the retention, migration, and backup of data sets; to control the retention and backup of aggregate groups, and to control the retention, backup, and class transition of objects.

manual tape library. A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which can be mounted on those drives. See also *tape library*. Contrast with *automated tape library*.

MEDIA2. Enhanced Capacity Cartridge System Tape

MEDIA3. High Performance Cartridge Tape

MEDIA4. Extended High Performance Cartridge Tape

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. If you wish to use the data set, it must be recalled. See also *migration level 1* and migration level 2.

migration control data set (MCDS). In DFSMShsm, a VSAM key-sequenced data set that contains statistics records, control records, user records, records for data sets that have migrated, and records for volumes under migration control of DFSMShsm.

migration level 1. DFSMShsm-owned DASD volumes that contain data sets migrated from primary storage volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 2*.

migration level 2. DFSMShsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 1*.

modified link pack area (MLPA). An area of virtual storage containing reenterable routines from the SYS1.LINKLIB, SYS1.SVCLIB, or SYS1.LPALIB system data sets that are to be part of the pageable extension of the link pack area during the current IPL. See also *link pack area*.

MVS configuration program (MVSCP). A single-step, batch program that defines the input/output configuration to MVS.

MVS/ESA. An MVS operating system environment that supports ESA/390.

MVS/ESA SP. An IBM licensed program used to control the MVS operating system. MVS/ESA SP together with DFSMS/MVS compose the base MVS/ESA operating environment. See also *OS/390*.

N

NaviQuest. A component of DFSMSdfp for implementing, verifying, and maintaining your DFSMS SMS environment in batch mode. It provides batch testing and reporting capabilities that can be used to automatically create test cases in bulk, run many other storage management tasks in batch mode, and use supplied ACS code fragments as models when creating your own ACS routines.

nonvolatile storage (NVS). Additional random access electronic storage with a backup battery power source, available with an IBM Cache Storage Control, used to retain data during a power outage. Nonvolatile storage, accessible from all storage directors, stores data during DASD fast write and dual copy operations.

O

OAM-managed volumes. Optical or tape volumes controlled by the object access method (OAM).

object. A named byte stream having no specific format or record orientation.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

OAM complex (OAMplex). One or more instances of OAM running on systems that are part of a parallel sysplex. The OAM systems that are part of an OAMplex share a common OAM database in a DB2 data-sharing group.

object backup storage group. A type of storage group that contains optical or tape volumes used for backup copies of objects. See also *storage group*.

object storage group. A type of storage group that contains objects on DASD, tape, or optical volumes. See also *storage group*.

object storage hierarchy. A hierarchy consisting of objects stored in DB2 table spaces on DASD, on optical or tape volumes that reside in a library, and on optical or tape volumes that reside on a shelf. See also *storage hierarchy*.

offline control data set (OCDS). In DFSMShsm, a VSAM key-sequenced set that contains information about tape backup volumes and tape migration level 2 volumes.

OpenEdition MVS. See *OS/390 UNIX System Services*

OpenEdition MVS file system. See *OS/390 UNIX file system*.

optical disk drive. The mechanism used to seek, read, and write data on an optical disk. An optical disk drive can be operator-accessible, such as the 3995 Optical Library Dataserver, or stand-alone, such as the 9346 or 9347 optical disk drives.

optical library. A storage device that houses optical drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives.

optical volume. Storage space on an optical disk, identified by a volume label. See also *volume*.

OS/390. OS/390 is a network computing-ready, integrated operating system consisting of more than 50 base elements and integrated optional features delivered as a configured, tested system. See also *MVS/ESA SP*.

OS/390 UNIX System Services (OS/390 UNIX). The set of functions provided by the SHELL and UTILITIES, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the OS/390 operating system that allow users to write and run application programs that conform to UNIX standards.

P

pageable link pack area (PLPA). An area of virtual storage containing SVC routines, access methods, and other read-only system and user programs that can be shared among users of the system. See also *link pack area*.

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

performance. (1) A measurement of the amount of work a product can produce with a given amount of resources. (2) In a DFSMS environment, a measurement of effective data processing speed with respect to objectives set by the storage administrator. Performance is largely determined by throughput, response time, and system availability.

permanent data set. A user-named data set that is normally retained for longer than the duration of a job or interactive session. Contrast with *temporary data set*.

physical storage. With respect to data, the actual space on a storage device that is to contain data.

pool storage group. A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also *storage group*.

primary data set. When referring to an entire data set collection, the primary data set is the first data set allocated. For individual data sets being stacked, the primary data set is the one in the data set collection that precedes the data set being stacked and is allocated closest to it.

primary storage. A DASD volume available to users for data allocation. The volumes in primary storage are called primary volumes. See also *storage hierarchy*. Contrast with *migration level 1* and *migration level 2*.

program management. The task of preparing programs for execution, storing the programs, load modules, or program objects in program libraries, and executing them on the operating system.

program object. All or part of a computer program in a form suitable for loading into virtual storage for execution. Program objects are stored in PDSE program libraries and have fewer restrictions than load modules. Program objects are produced by the binder.

R

recovery. The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a log.

Redundant Array of Independent Disks (RAID). A disk subsystem architecture that combines two or more physical disk storage devices into a single logical device to achieve data redundancy.

relative byte address (RBA). In VSAM, the displacement of a data record or a control interval from the beginning of the data set to which it belongs independent of the manner in which the data set is stored.

relative-record data set (RRDS). A VSAM data set whose records are loaded into fixed-length slots.

removable media library. The volumes that are available for immediate use, and the shelves where they could reside.

residence mode (RMODE). The attribute of a load module or program object

Resource Access Control Facility (RACF). An IBM-licensed program or a base element of OS/390, that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

Resource Measurement Facility (RMF). An IBM licensed program or optional element of OS/390, that measures selected areas of system activity and presents the data collected in the format of printed reports, system management facilities (SMF) records, or display reports. Use RMF to evaluate system performance and identify reasons for performance problems.

resource profile. A profile that provides RACF protection for one or more resources. User, group, and connect profiles are not resource profiles. The information in a resource profile can include the data set profile name, profile owner, universal access authority, access list, and other data. Resource profiles can be discrete profiles or generic profiles.

RLD count. The number of RLD records in a load module that follow the text block that the RLD count references. See also *RLD record*.

RLD record. A record in a relocation dictionary that contains information on relocatable address constants for that program object or load module.

S

server. (1) A function that provides services for users. (2) A machine that provides resources to the network. It provides a network service, such as disk storage and file transfer, or a program that uses such a service. See also *target*.

service level (Storage Management Subsystem). A set of logical characteristics of storage required by a Storage Management Subsystem-managed data set (for example, performance, security, availability).

service-level agreement. (1) An agreement between the storage administration group and a user group defining what service-levels the former will provide to ensure that users receive the space, availability, performance, and security they need. (2) An agreement between the storage administration group and operations defining what service-level operations will provide to ensure that storage management jobs required by the storage administration group are completed.

sharing control data set. A VSAM linear data set that contains information DFSMSdfp needs to ensure the integrity of the data sharing environment.

shelf. A place for storing removable media, such as tape and optical volumes, when they are not being written to or read.

shelf location. A single space on a shelf for storage of removable media.

small-data-set packing (SDSP). In DFSMShsm, the process used to migrate data sets that contain equal to or less than a specified amount of actual data. The data sets are written as one or more records into a VSAM data set on a migration level 1 volume.

SMS complex. A collection of systems or system groups that share a common configuration. All systems in an SMS complex share a common active control data set (ACDS) and a communications data set (COMMDS). The systems or system groups that share the configuration are defined to SMS in the SMS base configuration.

SMS control data set. A VSAM linear data set containing configurational, operational, or communications information that guides the execution of the Storage Management Subsystem. See also *source control data set*, *active control data set*, and *communications data set*.

source. That portion of the DDM architecture that is necessary for a system to provide source support, allowing a system providing this support to request access to remote data on a remote (target) system. See also *client*.

source control data set (SCDS). A VSAM linear data set containing an SMS configuration. The SMS configuration in an SCDS can be changed and validated using ISMF. See also *active control data set* and *communications data set*.

storage administration group. A centralized group within the data processing center that is responsible for managing the storage resources within an installation.

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage control. The component in a storage subsystem that handles interaction between processor channel and storage devices, runs channel commands, and controls storage devices.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD, optical, or tape volumes treated as a single object storage hierarchy. See also *VIO storage group, pool storage group, tape storage group, object storage group, object backup storage group*, and *dummy storage group*.

storage hierarchy. An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include main storage (memory, DASD cache), primary storage (DASD containing uncompressed data), migration level 1 (DASD containing data in a space-saving format), and migration level 2 (tape cartridges containing data in a space-saving format). See also *primary storage*, *migration level 1*, *migration level 2*, and *object storage hierarchy*.

storage location. A location physically separate from the removable media library where volumes are stored for disaster recovery, backup, and vital records management.

storage management. The activities of data set allocation, placement, monitoring, migration, backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automates these processes for you, while optimizing storage resources. See also *Storage Management Subsystem*.

Storage Management Subsystem (SMS). A DFSMS/MVS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

storage subsystem. A storage control and its attached storage devices. See also *tape subsystem*.

stripe. In DFSMS/MVS, the portion of a striped data set that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striped data set. In DFSMS/MVS, an extended-format data set consisting of two or more stripes. SMS determines the number of stripes to use based on the value of the SUSTAINED DATA RATE in the storage class. Striped data sets can take advantage of the sequential data striping access technique. See *striping* and *stripe*.

striping. A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

system data. The data sets required by MVS or its subsystems for initialization and control.

system group. All systems that are part of the same Parallel Sysplex and are running the Storage Management Subsystem with the same configuration, minus any systems in the Parallel Sysplex that are explicitly defined in the SMS configuration.

system-managed buffering for VSAM. A facility | available for system-managed extended-format VSAM | data sets in which DFSMSdfp determines the type of buffer management technique along with the number of buffers to use, based on data set and application specifications.

system-managed data set. A data set that has been assigned a storage class.

system-managed storage. Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, and space to applications. See also *DFSMS environment*.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See also *tape library*.

system-managed volume. A DASD, optical, or tape volume that belongs to a storage group. Contrast with *DFSMShsm-managed volume* and *DFSMSrmm-managed volume*.

system management facilities (SMF). A component of MVS that collects input/output (I/O) statistics, provided at the data set and storage class levels, which helps you monitor the performance of the direct access storage subsystem.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

T

tera (T). The information-industry meaning depends upon the context:

1. T = 1,099,511,627,776(2^{40}) for real and virtual storage
2. T = 1,000,000,000,000 for disk storage capacity (e.g. 4 Tb of DASD storage)
3. T = 1,000,000,000,000 for transmission rates

tape configuration database. One or more volume catalogs used to maintain records of system-managed tape libraries and tape volumes.

tape librarian. The person who manages the tape library.

tape library. A set of equipment and facilities that support an installation's tape environment. This can include tape storage racks, a set of tape drives, and a set of related tape volumes mounted on those drives. See also *system-managed tape library* and *automated tape library*.

Tape Library Dataserver. A hardware device that maintains the tape inventory associated with a set of tape drives. An automated tape library dataserver also manages the mounting, removal, and storage of tapes.

tape mount management. The methodology used to optimize tape subsystem operation and use, consisting of hardware and software facilities used to manage tape data efficiently.

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See also *storage group*.

tape subsystem. A magnetic tape subsystem consisting of a controller and devices, which allows for the storage of user data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

tape volume. A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

target. That portion of the DDM architecture that is necessary for a system to provide target support, allowing a system providing this support to receive and process requests from a remote (source) system. See also *server*.

temporary data set. An uncataloged data set whose name begins with & or &&, that is normally used only for the duration of a job or interactive session. Contrast with *permanent data set*.

threshold. A storage group attribute that controls the space usage on DASD volumes, as a percentage of occupied tracks versus total tracks. The *low migration threshold* is used during primary space management and interval migration to determine when to stop processing data. The *high allocation threshold* is used to determine candidate volumes for new data set allocations. Volumes with occupancy lower than the high threshold are selected over volumes that meet or exceed the high threshold value.

Transmission Control Protocol/Internet Protocol (TCP/IP). A suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network.

U

unit affinity. Requests that the system allocate different data sets residing on different removable volumes to the same device during execution of the step to reduce the total number of tape drives required to execute the step. Explicit unit affinity is specified by coding the UNIT=AFF JCL keyword on a DD statement. Implicit unit affinity exists when a DD statement requests more volumes than devices.

unit control block (UCB). A control block in storage that describes the characteristics of a particular I/O device on the operating system.

user group. A group of users in an installation who represent a single department or function within the organization.

V

validate. To check the completeness and consistency of an individual ACS routine or an entire SMS configuration.

virtual input/output (VIO) storage group. A type of storage group that allocates data sets to paging storage, which simulates a DASD volume. VIO storage groups do not contain any actual DASD volumes. See also *storage group*.

vital records. A data set or volume maintained for meeting an externally-imposed retention requirement, such as a legal requirement. Compare with *disaster recovery*.

vital record specification. Policies defined to manage the retention and movement of data sets and volumes for disaster recovery and vital records purposes.

volume. The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume*, *optical volume*, and *tape volume*.

volume mount analyzer. A program that helps you analyze your current tape environment. With tape mount management, you can identify data sets that can be redirected to the DASD buffer for management using SMS facilities.

volume status. In the Storage Management Subsystem, indicates whether the volume is fully available for system management:

- “Initial” indicates that the volume is not ready for system management because it contains data sets that are ineligible for system management.
- “Converted” indicates that all of the data sets on a volume have an associated storage class and are cataloged in an integrated catalog facility catalog.
- “Non-system-managed” indicates that the volume does not contain any system-managed data sets and has not been initialized as system-managed.

VSAM record-level sharing (VSAM RLS). An extension to VSAM that provides direct record-level sharing of VSAM data sets from multiple address spaces across multiple systems. Record-level sharing uses the System/390 Coupling Facility to provide cross-system locking, local buffer invalidation, and cross-system data caching.

VSAM sphere. The base cluster of a VSAM data set and its associated alternate indexes.

VSAM volume data set (VVDS). A data set that describes the characteristics of VSAM and system-managed data sets residing on a given DASD volume; part of an integrated catalog facility catalog. See also *basic catalog structure* and *integrated catalog facility catalog*.

Index

© Copyright IBM Corp. 1980, 1999

U.S.A

Readers' Comments — We'd Like to Hear from You

DFSMS/MVS Version 1 Release 5
General Information

Publication No. GC26-4900-06

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
GC26-4900-06

IBM®

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
RCF Processing Department
G26/050
5600 Cottle Road
SAN JOSE, CA 95193-0001

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

File Number: S370-20
Program Number: 5695-DF1
5647-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC26-4900-06

Spine information:

IBM DFSMS/MVS Version 1 Release 5 General Information